Residual Diffusion Bridge Model for Image Restoration

Hebaixu Wang[1,2], Jing Zhang[1†], Haoyang Chen[1,2], Haonan Guo[1,2], Di Wang[1,2], Jiayi Ma[1,2†], Bo Du[1,2†]
[1] Wuhan University, [2]Zhongguancun Academy
{wanghebaixu,jingzhang.cv,jyma2010}@gmail.com;{haoyangchen,haonan.guo,d_wang,dubo}@whu.edu.cn

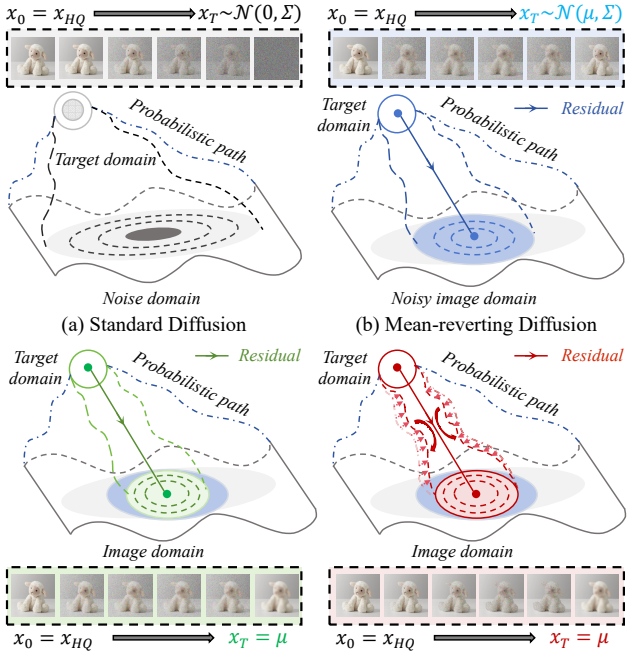
Abstract

Diffusion bridge models establish probabilistic paths between arbitrary paired distributions and exhibit great potential for universal image restoration. Most existing methods merely treat them as simple variants of stochastic interpolants, lacking a unified analytical perspective. Besides, they indiscriminately reconstruct images through global noise injection and removal, inevitably distorting undegraded regions due to imperfect reconstruction. To address these challenges, we propose the Residual Diffusion Bridge Model (RDBM). Specifically, we theoretically reformulate the stochastic differential equations of generalized diffusion bridge and derive the analytical formulas of its forward and reverse processes. Crucially, we leverage the residuals from given distributions to modulate the noise injection and removal, enabling adaptive restoration of degraded regions while preserving intact others. Moreover, we unravel the fundamental mathematical essence of existing bridge models, all of which are special cases of RDBM and empirically demonstrate the optimality of our proposed models. Extensive experiments are conducted to demonstrate the state-of-the-art performance of our method both qualitatively and quantitatively across diverse image restoration tasks. Code is publicly available at https://github.com/MiliLab/RDBM.

1. Introduction

Universal image restoration emerges as a unified paradigm integrating the perception, representation, and elimination of diverse degradations [18, 65], with the aim of restoring high-quality images from the degraded low-quality observation. It typically encompasses a broad range of classical tasks, including denoising, deraining, dehazing, super-resolving, and others [7, 13, 24, 63, 70]. Owing to the high fidelity of restored details, it has been widely adopted as a precursor across various downstream tasks [46, 81].

Diffusion models have achieved remarkable advances in universal image restoration [42]. Early methods [61, 71, 87] follow the standard diffusion paradigm [15, 22] that maps images to a Gaussian distribution, and initializes the reverse inference from pure noise. Some approaches leverage generative priors pretrained from large models [39, 45] as conditional guidance for denoising networks. Others treat various restoration tasks as inverse problems by assuming access to degradation kernels [8, 9, 64, 77]. However, the randomness of noise and reliance on specific priors compromise both stability and universality. Subsequent studies [36, 40] incorporate mean-reverting dynamics into dif-



(a) Standard Diffusion (b) Mean-reverting Diffusion (c) Diffusion Bridge (d) Ours

Figure 1. Typical diffusion processes for image restoration. (a) Standard diffusion maps high-quality images to the Gaussian noise domain. (b) Mean-reverting diffusion drives terminal state toward a low-quality domain with stationary noise. (c) Diffusion bridge establishes direct probabilistic transitions between known distributions. All inject noise globally, disrupting overall structures and constraining transitions. (d) In contrast, our RDBM selectively reconstructs degraded regions (e.g., the doll) while preserving intact areas (e.g., the background), thus avoiding redundant restoration.

†Corresponding author.

fusion stochastic differential equations (SDEs), clustering forward terminal states around degraded observations to retain task-relevant cues. Additionally, diffusion bridges [83] directly model point-to-point stochastic transitions between paired distributions, thereby strengthening data associations and improving restoration fidelity. Despite these advances, existing methods still rely on global noise perturbation to construct probabilistic trajectories, requiring rigid reverse denoising processes, as shown in Fig. 1. However, they fail to distinguish regions with varying degradation levels and imperfectly reconstruct intact regions, limiting restoration performance and adaptivity. Besides, a systematic and theoretical framework is absent to elucidate the intricate interconnections among existing diffusion bridge formulations.

In this work, we propose a scalable and unified diffusion bridge framework for universal image restoration, termed Residual Diffusion Bridge Model (RDBM), and conduct a comprehensive analysis of optimal distribution transitions. Specifically, we integrate the mean-reverting property of the Ornstein–Uhlenbeck SDEs with Doob's h-transform [57] to guide the terminal forward states toward degraded image distribution. Meanwhile, we use residuals from given distributions to dynamically modulate the probabilistic trajectories, thereby allowing the model to learn adaptive restoration of regions with varying degradation levels while mitigating redundant reconstruction in intact areas. Moreover, we theoretically demonstrate that our formulation yields the smooth distributional transition with respect to the residual-to-noise ratio. Building upon this, we uncover the mathematical essence of mainstream diffusion bridge formulations, all of which are special cases within our framework in specific configurations. Extensive experiments are conducted to verify the superiority of our method across diverse tasks including image restoration, translation, and inpainting both qualitatively and quantitatively. Our main contributions are summarized as follows:

1. We propose a scalable and unified diffusion bridge framework for image restoration. Theoretically, it is characterized as generalized stochastic interpolants that delineate probabilistic transitions between any paired distributions.
2. Benefiting from the certainty of terminal states, we exploit residuals from paired distributions to modulate noise injection and removal, enabling selective reconstruction of degraded regions while preserving intact areas.
3. We unify and reinterpret existing bridge models as special instances of our RDBM framework, and substantiate its generality and effectiveness through extensive theoretical analysis and empirical validation.

2. Related Work

Denoising diffusion models [61, 62] were initially developed for image generation. Methods such as DiffIR [71], DvSR [69], and SR3 [75] directly repurpose diffusion models conditioned on degraded images for image restoration task, suffering from performance bottlenecks for task incompatibility. I2SB [34] and ColdDiffusion [5] bypass explicit noise perturbations and instead learn a degraded diffusion process directly via the network. Besides, RDDM [36], ReShift [74], ResFusion [58], and DiffUIR [82] incorporate degradation distributions into the perturbation kernels to explicitly characterize degradation-aware diffusion processes. Moreover, IRSDE [40] employs a mean-reverting process to enforce diffusion trajectories that regress toward noisy degraded images with stationary variance. Additionally, DDBM [83], BBDM [29] and GOUB [73] further apply Doob's h-transform to remove terminal noise, offering a tractable alternative to pave the probability path that connects degraded and clean images, thereby achieving remarkable restoration performance. Flow matching [32, 37] discards the stochastic noise and constructs the deterministic distribution transition path, thereby facilitating the optimal transport [86]. In contrast, our RDBM introduces the residual to modulate noise perturbation, enabling spatially adaptive restoration. Besides, RDBM can extend to these diffusion bridge models and flow matching in specific settings.

3. Background

3.1. Diffusion Bridge Models

Diffusion SDEs [15, 60] with drift term $\mathbf{f}(\cdot, t)$ and diffusion term $g(t)$ can be generally formulated as [23]:

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}_t, t)dt + g(t)d\omega_t, \tag{1}$$

where ω_t is the standard Wiener process. Eq. (1) describes the stochastic process from initial data $\mathbf{x}_0 \sim p_{data}(\mathbf{x})$ to a prior distribution $\mathbf{x}_T \sim p_{prior}(\mathbf{x})$. Its reverse SDEs and probability flow ordinary differential equations (ODEs) that share the same marginal distributions can be derived as:

$$d\mathbf{x}_t = [\mathbf{f}(\mathbf{x}_t, t) - g^2(t)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)]dt + g(t)d\overline{\omega}_t, \tag{2}$$

$$d\mathbf{x}_t = [\mathbf{f}(\mathbf{x}_t, t) - \frac{1}{2}g^2(t)\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)]dt, \tag{3}$$

where $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_t)$ is score function. Furthermore, a diffusion process defined in Eq. (1) can be driven to arrive at a particular point of interest $\boldsymbol{\mu}$ via Doob's h-transform [55]:

$$d\mathbf{x}_t = [\mathbf{f}(\mathbf{x}_t, t) + g(t)^2\mathbf{h}(\mathbf{x}_t, t, \mathbf{x}_T, T)]dt + g(t)d\omega_t, \tag{4}$$

where $\mathbf{h}(\mathbf{x}_t, t, \mathbf{x}_T, T) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_T|\mathbf{x}_t)$ is the gradient of the log transition kernel from t to T generated by the original SDE. When both the initial state $\mathbf{x}_0$ and terminal state $\mathbf{x}_T = \boldsymbol{\mu}$ are fixed, Eq. (4) defines a stochastic process known as a diffusion bridge (see proof in Suppl. A).

3.2. Ornstein Uhlenbeck Process

Ornstein–Uhlenbeck (OU) process is a stationary Gaussian-Markov process, with its marginal distribution converging

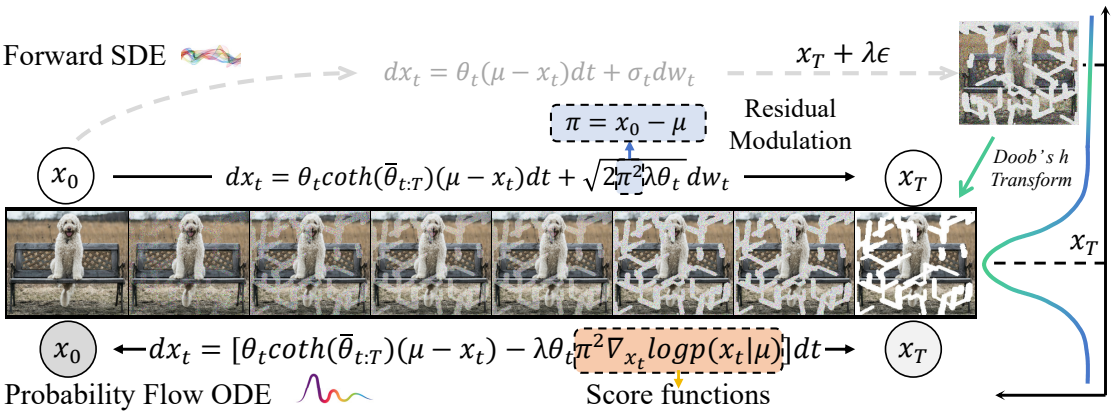


Figure 2. A schematic of Residual Diffusion Bridge Models. RDBM utilizes a diffusion process guided by Doob's h-transform towards an endpoint $\mathbf{x}_T = \boldsymbol{\mu}$ free from stationary noise $\lambda\epsilon$. Modulated by the residual component $\boldsymbol{\pi} = \mathbf{x}_0 - \mathbf{x}_T$, the noise perturbation is selectively imposed on different regions with diverse degradation levels, thereby constructing probabilistic paths. Besides, it learns to reverse the process by matching the residual bridge score functions, facilitating an adaptive inversion from $\mathbf{x}_T \sim p_{prior}(\boldsymbol{x})$ to $\mathbf{x}_0 \sim p_{data}(\boldsymbol{x})$.

toward a stable mean $\boldsymbol{\mu}$ with fixed variance over time. Formally, the OU process is generally defined as follows:

$$d\mathbf{x}_t = \theta_t(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sigma_t d\omega_t, \tag{5}$$

where θ_t and σ_t respectively denote time-dependent drift and diffusion coefficients that characterize the speed of the mean-reversion. The transition probability of Eq. (5) admits a closed-form solution as below:

$$p(\mathbf{x}_t \mid \mathbf{x}_s) = \mathcal{N}(\bar{\mathbf{m}}_{s:t}, \bar{\sigma}^2_{s:t}\boldsymbol{I}) =, \tag{6}$$

$$\mathcal{N}(\boldsymbol{\mu} + (\mathbf{x}_s - \boldsymbol{\mu})e^{-\bar{\theta}_{s:t}}, \int_s^t \sigma_z^2 e^{-2\bar{\theta}_{z:t}} dz) \tag{7}$$

$$\bar{\theta}_{s:t} = \int_s^t \theta_z dz. \tag{8}$$

Driven by the mean-reverting dynamics with Gaussian perturbations, the diffusion trajectory originates from $\mathbf{x}_0 \sim p_{data}(\mathbf{x})$ at initial time and gradually approaches $\mathbf{x}_T = \boldsymbol{\mu} \sim p_{prior}(\mathbf{x})$ at final time T. See Suppl. B for details.

4. Residual Diffusion Bridge Models

4.1. Generalized Forward Process

We redefine a OU process in Eq. (5) for generality:

$$d\mathbf{x}_t = \theta_t(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t d\omega_t, \tag{9}$$

where $\boldsymbol{\pi}$ is a predefined value. By applying the Doob's h-transform to Eq. (9), we can establish the diffusion bridge that connects the high-quality image distribution $\mathbf{x}_0 \sim p_{HQ}(\boldsymbol{x})$ with degraded image distribution $\boldsymbol{\mu} \sim p_{LQ}(\boldsymbol{x})$:

Proposition 1 *Let $\mathbf{x}_t$ be a finite random variable governed by the generalized diffusion bridge process in Eq. (9), with terminal condition $\mathbf{x}_T = \boldsymbol{\mu}$. The evolution of its marginal distribution $p(\mathbf{x}_t \mid \mathbf{x}_T)$ satisfies the following SDE under a fixed drift-to-diffusion coefficient ratio $\lambda = \sigma_t^2/(2\theta_t)$:*

$$d\mathbf{x}_t = \theta_t \coth(\bar{\theta}_{t:T})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sqrt{2\boldsymbol{\pi}^2\lambda\theta_t}d\omega_t, \tag{10}$$

where $\bar{\theta}_{s:t} = \int_s^t \theta_z dz$ and $\boldsymbol{\pi} \in \mathbb{R}$. See Suppl. C.

Consequently, Eq. (10) describes the generalized diffusion bridge models governed by λ, θ_t and $\boldsymbol{\pi}$. Here, λ controls the global noise level, while θ_t and $\boldsymbol{\pi}$ jointly determine bridge category and dynamical evolution. Furthermore, we can derive its closed-form solution as follows:

Proposition 2 *Given an initial state $\mathbf{x}_0$, the analytical solution of $\mathbf{x}_t$ at time $0 < t < T$ of that SDE in Eq. (10) can be formulated as:*

$$\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\frac{\sinh(\bar{\theta}_{t:T})}{\sinh(\bar{\theta}_{0:T})} + \int_0^t \sqrt{2\boldsymbol{\pi}^2\lambda\theta_s}\frac{\sinh(\bar{\theta}_{t:T})}{\sinh(\bar{\theta}_{s:T})}d\omega_s, \tag{11}$$

which satisfies a Gaussian distribution with expectation $E[\mathbf{x}_t]$ and variance $Var[\mathbf{x}_t]$ (proof is provided in Suppl. C):

$$E[x_t] = \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\frac{\sinh(\bar{\theta}_{t:T})}{\sinh(\bar{\theta}_{0:T})} := \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_t, \tag{12}$$

$$Var[x_t] = 2\boldsymbol{\pi}^2\lambda\frac{\sinh(\bar{\theta}_{0:t})\sinh(\bar{\theta}_{t:T})}{\sinh(\bar{\theta}_{0:T})} := \boldsymbol{\pi}^2\Sigma_t^2. \tag{13}$$

Eq. (11) unveils that the trajectory of probability is dictated by a weighted amalgamation of the residual and Gaussian noise. In order to delineate its temporal dynamic evolution, we define the residual-to-noise ratio $R(t, i, j)$ for each pixel i, j at time t as follows (details are in Suppl. D):

$$R(i,j,t) = \frac{[x_0(i,j) - \boldsymbol{\mu}(i,j)]^2}{2[\boldsymbol{\pi}(i,j)]^2\lambda}\frac{\sinh(\bar{\theta}_{t:T})}{\sinh(\bar{\theta}_{0:t})\sinh(\bar{\theta}_{0:T})}, \tag{14}$$

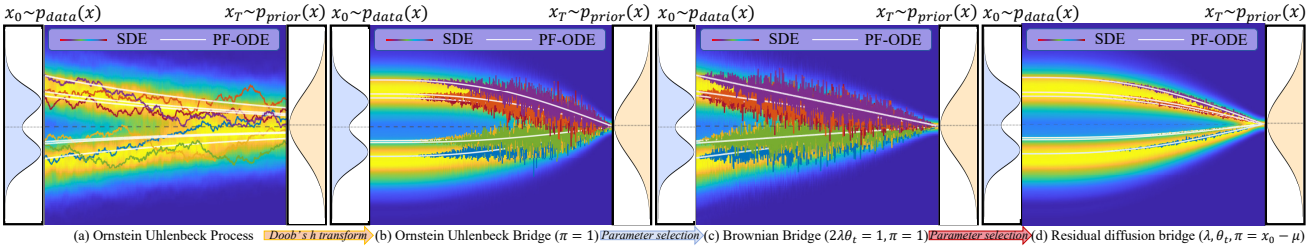


Figure 3. Overview of mainstream diffusion processes via SDEs, all of which are special cases of our framework. (a) OU process maps the data distribution to the prior distribution with noise. (b) OU bridge constructs probabilistic transition paths between given distributions. (c) Brownian bridge models linear expectations of intermediate states. (d) Our RDBM leverages residuals from paired distributions to adaptively modify the transition trajectories, maintaining a smooth residual-to-noise ratio.

which is governed by two terms. The first term depends on the residual component and fixed ratio λ. The second term is entirely determined by θ_t series and exhibits a monotonic decline, which diverges to infinity as time $t \to 0$ and converges to an infinitesimal value as time $t \to T$. Previous works [40, 73, 83] typically set $\boldsymbol{\pi} = 1$, thereby performing the global noise perturbation to uniformly disrupt the overall structure of images. This induces two ill-posed issues: (i) degraded regions with varying levels are treated equally and intact regions suffer redundant and imperfect reconstruction due to inevitable cumulative error in reverse process. (ii) pixel-wise numerator $[x_0(i,j) - \boldsymbol{\mu}(i,j)]^2$ may exhibit discontinuous jumps, potentially distorting the smooth monotonic decay of the residual-to-noise ratio. Therefore, to maintain the dynamic equilibrium in transmission trajectories, we fix $\boldsymbol{\pi} = \mathbf{x}_0 - \boldsymbol{\mu}$, thereby deriving our specific formulation within this framework with adaptive noise perturbation and pixel-independent residual-to-noise ratio:

$$R(t,i,j) = R(t) \propto \frac{\sinh(\overline{\theta}_{t:T})}{\sinh(\overline{\theta}_{0:t})\sinh(\overline{\theta}_{0:T})}. \tag{15}$$

4.2. Reverse Process and Training Objective

From Eq. (12)-(13), the transition probability distributions from initial state $\mathbf{x}_0$ to intermediate states $\mathbf{x}_t$ and $\mathbf{x}_{t-1}$ are:

$$q(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_t, \boldsymbol{\pi}^2\Sigma_t^2\boldsymbol{I}), \tag{16}$$

$$q(\mathbf{x}_{t-1}|\mathbf{x}_0,\boldsymbol{\mu}) = \mathcal{N}(\boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_{t-1}, \boldsymbol{\pi}^2\Sigma_{t-1}^2\boldsymbol{I}), \tag{17}$$

Supposing that sampling from $\mathbf{x}_t$ to $\mathbf{x}_{t-1}$ follows the Gaussian distribution, we leverage *Bayes' theorem* to derive the deterministic sampling of reverse process (see Suppl. E):

$$\mathbf{x}_{t-1} = \boldsymbol{\mu} + \frac{\Sigma_{t-1}}{\Sigma_t}(\mathbf{x}_t - \boldsymbol{\mu}) + (\Theta_{t-1} - \Theta_t\frac{\Sigma_{t-1}}{\Sigma_t})\boldsymbol{\pi}, \tag{18}$$

$$= \boldsymbol{\mu} + \frac{\Theta_{t-1}}{\Theta_t}(\mathbf{x}_t - \boldsymbol{\mu}) - (\frac{\Theta_{t-1}}{\Theta_t}\Sigma_t - \Sigma_{t-1})\boldsymbol{\pi}\epsilon_t, \tag{19}$$

Apparently, Eq. (19) involves two unknowns, the residual $\boldsymbol{\pi}$ and the noise ϵ_t. In theory, the distributions at all time steps should be aligned; thus, the overall training objective is:

$$\mathcal{L}(\dot{\theta}) = D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t,\mathbf{x}_0,\boldsymbol{\mu})||p_{\dot{\theta}}(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{\mu})). \tag{20}$$

Table 1. Connections to other mainstream bridge models.

Diffusion Bridge Configurations			Method
$\theta_t \to 0$	λ	$\boldsymbol{\pi} = 0$	Flow Matching [32, 37]
$\theta_t \to 0$	$\lambda \to \infty$	$\boldsymbol{\pi} = 1$	VE Bridge [83]
$\theta_t \to 0$	$\lambda \to \frac{1}{2}$	$\boldsymbol{\pi} = 1$	VP Bridge [83]
$\theta_t \to 0$	$2\lambda\theta_t \to 1$	$\boldsymbol{\pi} = 1$	Brownian Bridge [29, 34]
θ_t	λ	$\boldsymbol{\pi} = 1$	OU Bridge [73]
θ_t	λ	$\boldsymbol{\pi} = \mathbf{x}_0 - \boldsymbol{\mu}$	Ours

Assuming $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t,\boldsymbol{\mu})$ follows a Gaussian distribution centered at $m_{\dot{\theta}}(\mathbf{x}_t,\mathbf{x}_0)$ with a constant variance, minimizing the Kullback-Leibler divergence [25] D_{KL} is equivalent to reducing the distance between the means (see Suppl. G):

$$\mathcal{L}(\dot{\theta}) := \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x}_0)}[\eta_m\|m(\mathbf{x}_t,\mathbf{x}_0) - m_{\dot{\theta}}(\mathbf{x}_t,\mathbf{x}_0)\|] \tag{21}$$

$$:= \mathbb{E}_{\mathbf{x}_0,\boldsymbol{\mu},t}[\eta_\epsilon\|\boldsymbol{\pi}_\epsilon^{\dot{\theta}}(\mathbf{x}_t,t,\boldsymbol{\mu}) - (\mathbf{x}_0 - \boldsymbol{\mu})\epsilon_t\|], \tag{22}$$

where η_m and η_ϵ are different weights for different training objectives. Accordingly, we can employ a neural network $\boldsymbol{\pi}_\epsilon^{\dot{\theta}}(\mathbf{x}_t,t,\boldsymbol{\mu})$ to predict the multiplication of residual and noise at once. The detailed algorithms for training and sampling are presented in Alg. 1 and Alg. 2, respectively.

4.3. Analysis

We redefine a general mean-reverting process in Eq. (9) and employ Doob's h transform to derive the generalized diffusion bridge in Eq. (10) that exhibits the property of mean-arrival. Visualization comparisons of probability paths with several diffusion processes are illustrated in Fig. 3. We configure $\boldsymbol{\pi}$ to serve as the residual component for adaptive noise perturbation, yielding a smoothly decaying residual-to-noise ratio highly compatible with image restoration. Besides, other mainstream bridge models can be concluded in our framework, such as standard diffusion bridge [83], Brownian Bridge [29, 34], OU Bridge [73], Flow Matching [32, 37] and others, as summarized in Tab. 1. Please see Suppl. F for detailed derivations.

Algorithm 1: Training.

Input: Clean image $\mathbf{x}_0$;
Degraded image: $\boldsymbol{\mu}$;
Residual map: $\boldsymbol{\pi} = \mathbf{x}_0 - \boldsymbol{\mu}$.

repeat
  $\mathbf{x}_0 \sim q(\mathbf{x}_0)$;
  $t \sim Uniform(1, \cdots, T)$;
  $\epsilon \sim \mathcal{N}(0, \boldsymbol{I})$;
  $\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_t + \boldsymbol{\pi}\Sigma_t\epsilon$;
  Take the gradient descent step on
    $\nabla_\theta \|\boldsymbol{\pi}\epsilon - \boldsymbol{\pi}_\epsilon^{\dot{\theta}}(\mathbf{x}_t, t, \boldsymbol{\mu})\|_1$
until *converged*;

Algorithm 2: Sampling.

Input: Degraded image: $\boldsymbol{\mu}$;
Neural network $\boldsymbol{\pi}_\epsilon^\theta(\cdot)$.

for $t = T$ to 1 **do**
  $\boldsymbol{\pi}\epsilon = \boldsymbol{\pi}_\epsilon^\theta(x_t, t, \boldsymbol{\mu})$
  if $t = T$ **then**
    $\mathbf{x}_T = \boldsymbol{\mu}$
  else
    $\mathbf{x}_{t-1} = \boldsymbol{\mu} + \frac{\Theta_{t-1}}{\Theta_t}(\mathbf{x}_t - \boldsymbol{\mu}) - (\frac{\Theta_{t-1}}{\Theta_t}\Sigma_t - \Sigma_{t-1})\boldsymbol{\pi}\epsilon$
end
Output: $\mathbf{x}_0$.

5. Experiments

5.1. Datasets and Evaluation Metrics

Extensive experiments are conducted to assess the performance of our method on five image restoration tasks, including deraining, low-light enhancement, desnowing, dehazing, and deblurring. For fairness, we collect and mix the most widely used datasets for each task as follows. Besides, dataset details are as summarized in Suppl. H.

Image deraining. We train our model on the merged datasets from Rain13K [19] and DeRaindrop [51], which cover diverse rain streaks and densities. Evaluation is conducted on both rain- and raindrop-removal tasks using the mixed datasets [31, 51, 72, 78]. In addition, we assess zero-shot generalization on real-world datasets, including GT-Rain [4] without ground-truth for reference.

Low-light enhancement. We combine the LOL [68] and VE-LOL-L [35] datasets, which furnish real and synthetic paired samples across diverse scenes with varying illumination and noise levels. Additionally, we employ the NPE [66], MEF [44] and DICM [26] datasets to conduct zero-shot generalization on real-world scenarios.

Image desnowing. We adopt the CSD [6] dataset as the primary benchmark and evaluate real-world performance on Snow100K-Real [38], which has no ground-truth.

Image dehazing. We adopt ITS_v2 [27] and D-HAZY [10] as training benchmarks, encompassing diverse scenes under varying haze densities. The outdoor subset SOTS [27] is used for evaluation, while real-world generalization is assessed on Dense-Haze [2], NHRW [79], and NH-HAZE [3].

Image deblurring. We use the GoPro [48] dataset to perform deblurring tasks, which contains various levels of blur obtained by averaging the clear images captured in very short intervals. To further validate the generalizability, we perform zero-shot restoration on the RealBlur [53] dataset.

Benchmarks are evaluated using peak signal-to-noise ratio (PSNR) [16], structural similarity (SSIM) [67], natural image quality evaluator (NIQE) [26] in RGB space, and the learned perceptual image patch similarity (LPIPS) [59] in feature space. For fairness, we compare our method with several universal restoration methods [11, 12, 20, 28, 30, 40, 41, 43, 50, 52, 73, 76, 80], which are all re-implemented on the mixed datasets for comparisons.

5.2. Implementation Details

Our method is trained using 8 Nvidia A800 GPUs with PyTorch [49] framework for 128h. Adam optimizer and L1 loss are employed for 500k iterations with a learning rate of $1e{-}4$. We set the batch size as 20 and distribute it evenly to each task. We randomly crop patches of size 256×256 from the original image as network input for training and use 10 timesteps for full-resolution testing. We utilize U-Net [56] architecture as network backbone. We change the channel number of the hidden layers C to obtain different versions with varied parameter quantities:

- RDBM-T: C=32, channel multiplier = {1,1,1,1}
- RDBM-S: C=32, channel multiplier = {1,2,2,4}
- RDBM-B: C=64, channel multiplier = {1,2,2,4}
- RDBM-L: C=64, channel multiplier = {1,2,4,8}

5.3. Comparative Experiments

We compare our RDBM with several representative universal methods across five challenge image restoration tasks.

Visual comparison. The qualitative results are illustrated in Fig. 4. For more results, please refer to Suppl. H. Obviously, our method generates high-quality results that are the most similar to ground-truth compared with other methods.

Quantitative evaluation. We present quantitative results in Tab. 2. Clearly, RDBM-L attains great performance improvement across all tasks by a large margin, culminating in average gains of 1.55 dB in PSNR. For fairness, we also evaluate several lightweight RDBM variants. Notably, RDBM-B also gets the best average PSNR and SSIM with fewer parameters than competing methods, highlighting the effectiveness of our design. Moreover, our models exhibit high scalability across different parameter levels. In conclusion, our method is the most competitive.

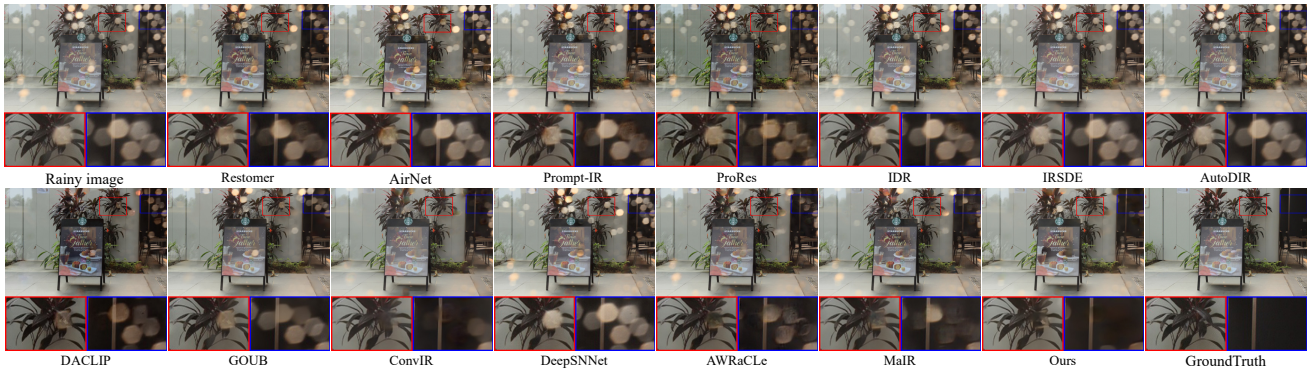


Figure 4. Visualization comparison with state-of-the-art methods on deraining. Zoom in for best view.

Table 2. Quantitative comparisons of five image restoration tasks. The FLOPS is calculated in the inference stage with 256×256 resolution. The best and second best results of universal models are shown in **red** and **blue**, respectively.

Method	Year	Deraining		Enhancement		Desnowing		Dehazing		Deblurring		Average		Complexity	
		PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	Params(M)	FLOPs(G)
Restomer [76]	2022	28.54	0.847	21.75	0.742	28.53	0.919	26.54	0.924	26.44	0.799	27.61	0.869	26.09	140.99
AirNet [28]	2022	24.78	0.774	13.05	0.485	25.80	0.885	18.53	0.827	25.76	0.782	24.01	0.809	5.76	301.27
Prompt-IR [50]	2023	28.97	0.856	20.97	0.733	29.52	0.938	25.80	0.929	26.25	0.797	27.89	0.878	32.96	158.14
ProRes [43]	2023	22.42	0.752	20.31	0.741	24.53	0.859	24.81	0.888	26.08	0.792	24.08	0.814	370.26	97.17
IDR [80]	2023	28.40	0.844	20.95	0.706	27.77	0.911	24.48	0.914	26.33	0.799	26.96	0.863	6.19	32.16
IRSDE [40]	2023	24.05	0.822	11.29	0.450	15.91	0.806	11.52	0.697	26.68	0.811	19.55	0.783	137.13	379.33
AutoDIR [20]	2024	29.32	0.863	15.65	0.707	15.31	0.706	19.01	0.829	**28.47**	**0.864**	22.43	0.799	115.86	63.38
DA-CLIP [41]	2024	28.63	0.854	19.50	0.730	28.23	0.934	27.26	0.941	26.47	0.818	27.54	0.881	32.96	158.14
GOUB [73]	2024	28.65	**0.870**	17.80	0.723	30.39	**0.960**	20.85	0.902	27.85	0.838	27.60	0.895	137.13	379.34
ConvIR [11]	2024	29.18	0.867	21.36	**0.771**	**31.43**	0.950	29.13	0.960	28.41	0.862	**29.49**	0.903	14.82	128.93
DeepSNNet [12]	2025	28.62	0.845	17.90	0.661	30.02	0.927	28.72	0.937	25.81	0.773	28.15	0.865	17.32	71.79
AWRaCLe [52]	2025	29.15	0.860	20.41	0.756	27.70	0.927	18.38	0.789	26.37	0.818	26.31	0.861	94.18	165.42
MaIR [30]	2025	**29.45**	0.864	**21.76**	0.750	30.80	0.955	**30.39**	**0.960**	28.28	0.859	29.51	**0.904**	20.71	110.44
RDBM-T	-	27.98	0.844	21.04	0.745	28.47	0.918	26.88	0.928	25.82	0.784	27.31	0.865	0.45	5.74
RDBM-S	-	29.23	0.864	21.98	0.765	30.93	0.941	28.92	0.942	26.67	0.808	28.99	0.886	1.07	8.01
RDBM-B	-	29.70	0.875	22.00	0.761	32.48	0.956	31.56	0.966	27.81	0.842	30.24	0.904	3.65	23.97
RDBM-L	-	**30.31**	**0.884**	**24.53**	**0.812**	**32.59**	**0.961**	**33.45**	**0.965**	**29.04**	**0.877**	**31.04**	**0.917**	7.73	32.93

Table 3. Performance of different noise schedule ($\lambda = 10/255$).

Schedule	Deraining		Enlighening		Desnowing		Dehazing		Deblur		Average	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
Linear	29.63	0.878	22.39	0.774	34.15	0.965	32.00	0.958	28.48	0.864	30.99	0.912
Cosine	30.31	0.884	24.53	0.812	32.59	0.961	33.45	0.965	29.04	0.877	31.04	0.917
Sigmoid	29.31	0.868	22.91	0.782	33.80	0.961	32.06	0.970	28.63	0.869	30.84	0.911

Table 4. Performance of varied stationary variance λ.

λ	Deraining		Enlighening		Desnowing		Dehazing		Deblur		Average	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
1/255	30.03	0.884	23.87	0.823	31.86	0.957	31.27	0.959	28.89	0.874	30.36	0.915
10/255	30.31	0.884	24.53	0.812	32.59	0.961	33.45	0.965	29.04	0.877	31.04	0.917
20/255	29.86	0.879	22.20	0.756	33.40	0.962	31.52	0.966	28.08	0.850	30.66	0.909
50/255	29.94	0.884	22.61	0.782	32.66	0.964	29.51	0.950	28.55	0.867	30.26	0.913
100/255	29.98	0.880	23.49	0.794	30.09	0.950	27.03	0.940	28.53	0.865	29.08	0.906

5.4. Ablation Study

To thoroughly explore the efficacy of our method, we carry out ablation studies encompassing three distinct categories: **Influence of various implementation configurations.** Our RDBM formulations are governed by the schedule $\{\theta_t\}$ and stationary variance λ. Initially, we adopt the empirical choice $\lambda = \frac{10}{255}$ [73] and compare performance across different noise schedules, as reported in Tab. 3. It is evident that the optimal noise schedules differ for distinct restoration tasks, with the cosine schedule generally yielding the best results. Building on this finding, we further conduct a quantitative comparison among diverse stationary variance λ, as presented in Tab. 4. The results indicate that $\lambda = \frac{10}{255}$ with cosine noise schedule is the optimal configuration.

Performance across different sampling steps. Model efficiency and restoration quality hinge on the sampling steps, quantified by neural function evaluations (NFEs). We provide the restoration performance of different sampling steps in Tab. 5. Clearly, our model exhibits varying performance across different NFEs. Initially, the restoration performance increases with more steps and peaks at 10 NFEs, reflecting accuracy gains from additional iterations. Beyond this threshold, performance gradually declines as NFEs rise. The underlying reasons are that our model is designed to handle diverse degradation types within a unified framework. In scenarios where samples exhibit multiple degradations, the model tends to prioritize the removal of the primary degradation before addressing secondary ones. Con-

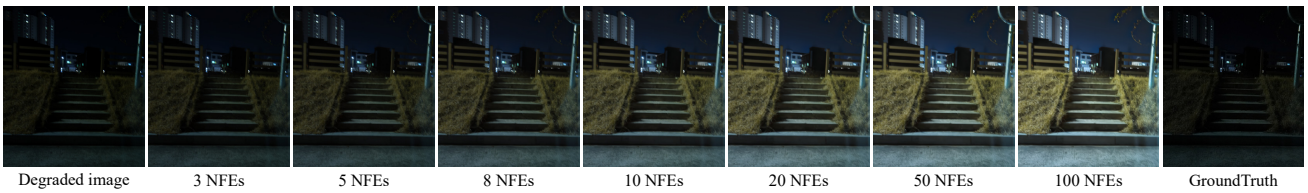


Figure 5. Visualization results of different NFEs in a blurry night-time scene. Zoom in for best view.

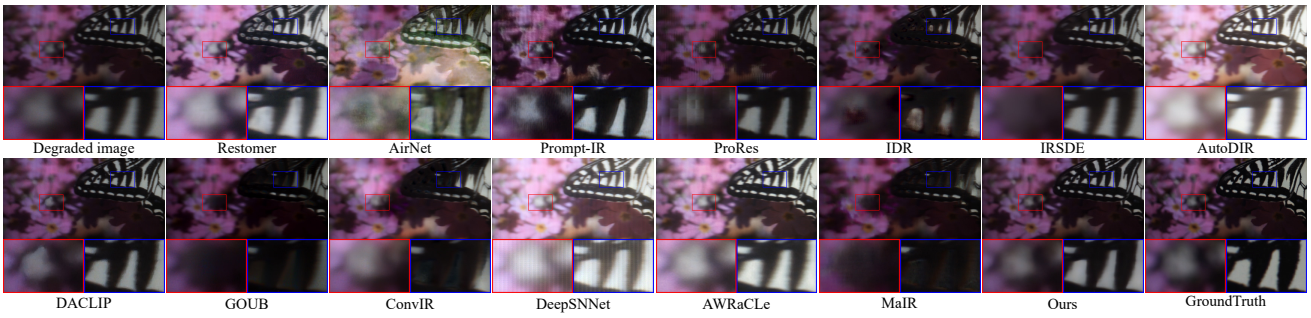


Figure 6. Visualization results of zero-shot generalization in real-world TOLED dataset. Zoom in for best view.

Table 5. Restoration performance of different sampling steps.

NFE	Deraining		Enlighening		Desnowing		Dehazing		Deblur		Average	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
2	26.06	0.790	14.82	0.648	20.28	0.835	17.20	0.809	28.06	0.855	22.81	0.815
5	30.05	0.876	23.35	0.809	30.47	0.947	29.01	0.929	29.13	0.878	29.61	0.905
10	30.31	0.884	24.53	0.812	32.60	0.961	33.45	0.965	29.04	0.877	31.04	0.917
20	30.10	0.882	24.35	0.811	31.96	0.959	32.25	0.961	28.94	0.875	30.58	0.915
50	29.92	0.880	24.21	0.809	31.59	0.958	31.56	0.958	28.85	0.873	30.28	0.913
100	29.84	0.879	24.13	0.808	31.49	0.957	31.39	0.957	28.80	0.873	30.19	0.912

Table 6. Restoration performance of different π.

π	Deraining		Enlighening		Desnowing		Dehazing		Deblur		Average	
	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑	PSNR↑	SSIM↑
0	28.10	0.841	19.68	0.722	30.24	0.927	28.13	0.936	26.58	0.806	28.21	0.872
1	29.56	0.876	21.71	0.749	32.79	0.957	30.74	0.961	27.66	0.838	30.15	0.903
$x_0 - x_T$	30.31	0.884	24.53	0.812	32.59	0.961	33.45	0.965	29.04	0.877	31.04	0.917
$\lvert x_0 - x_T \rvert$	30.36	0.883	24.37	0.812	32.40	0.957	33.19	0.965	28.99	0.876	30.94	0.915

sequently, the restored output may deviate from the available reference, as shown in Fig 5. In conclusion, we adopt 10 sampling steps to ensure performance and efficiency.

Impact of diverse diffusion bridge settings. By appropriately selecting $\boldsymbol{\pi}$, our method can establish equivalence with other diffusion bridges. Hence, we perform the restoration performance comparisons with different $\boldsymbol{\pi}$ selections, as presented in Tab. 6. The model is akin to flow matching as $\boldsymbol{\pi} = 0$, yielding moderate results. It resembles stochastic interpolants and performs better as $\boldsymbol{\pi} = 1$. Configuring $\boldsymbol{\pi}$ as the distributional residual or its absolute value is our formulation. These two variants produce similar results and achieve the best overall performance, thus verifying that residual bridge score matching offers a robust and effective paradigm for universal image restoration.

5.5. Zero-Short Real-world Generation

To evaluate the generalization ability of our method, we do zero-shot generalization for unknown and known restoration tasks in real-world scenes. "Unknown" denotes cases where the degradation type is unspecified and may be compound, whereas "known" matches our task specification. As all methods are re-implemented on mixed datasets, they inherently handle diverse degradation types. In comparison, our method achieves strong performance in both settings.

Table 7. Comparison under unknown tasks setting (under-display camera image restoration) on POLED and TOLED datasets.

Method	POLED				TOLED			
	PSNR↑	SSIM↑	MSE↓	LPIPS↓	PSNR↑	SSIM↑	MSE↓	LPIPS↓
Restomer [76]	11.500	0.445	0.077	0.494	11.094	0.495	0.106	0.330
AirNet [28]	5.705	0.103	0.324	1.072	9.706	0.430	0.117	0.403
Prompt-IR [50]	11.589	0.429	0.075	0.541	13.088	0.504	0.105	0.336
ProRes [43]	10.284	0.433	0.103	0.473	**28.452**	**0.834**	**0.002**	0.212
IDR [80]	13.583	0.466	0.057	0.551	24.259	0.759	0.008	0.253
IRSDE [40]	**16.983**	**0.615**	0.029	0.475	27.163	0.811	0.002	0.243
AutoDIR [20]	8.627	0.404	0.151	**0.406**	9.354	0.443	0.130	0.338
DA-CLIP [41]	16.788	0.559	**0.025**	0.469	27.256	0.789	0.003	**0.201**
GOUB [73]	12.922	0.525	0.053	0.446	23.177	0.761	0.007	0.269
ConvIR [11]	9.370	0.429	0.130	0.477	13.659	0.558	0.091	0.316
DeepSNNet [12]	10.195	0.411	0.113	0.534	17.394	0.576	0.075	0.303
AWRaCLe [52]	11.208	0.431	0.091	0.513	10.540	0.495	0.121	0.331
MaIR [30]	11.072	0.423	0.086	0.529	23.637	0.770	0.005	0.271
RDBM	**19.834**	**0.715**	**0.012**	**0.351**	**30.809**	**0.870**	**0.001**	**0.202**

Unknown task generalization. POLED and TOLED [84] are captured by under-display cameras in high-resolution with different degradation types, which fully meet the real-world scene. The quantitative results are reported in Tab. 7 while the visual comparisons are illustrated in Fig. 6. Evidently, our method achieves the best metric evaluation and our restored image is the most similar to ground-truth.

Known task generalization. As real-world datasets mainly have no ground truth, we use the non-reference metric, i.e., MetaIQA [85] and NIQE [47], to assess the perceptual quality, as provided in Tab. 8. Results show that our method outperforms other universal models in various benchmarks.

Figure 7. Visualization results of image translation (top row) and image inpainting (bottom row). Zoom in for best view.

Table 8. Comparison under known task generalization setting.

Method	Deraining		Enhancement		Desnowing		Dehazing		Deblurring	
	MetalQA↑	NIQE↓	MetalQA↑	NIQE↓	MetalQA↑	NIQE↓	MetalQA↑	NIQE↓	MetalQA↑	NIQE↓
Restomer [76]	0.231	13.115	0.328	3.828	0.357	5.845	0.437	4.400	0.303	6.734
AirNet [28]	0.232	11.668	0.280	3.674	0.347	6.091	0.440	4.623	0.286	6.393
Prompt-IR [50]	0.232	11.439	0.308	3.797	0.361	5.840	0.437	4.962	0.286	6.670
ProRes [43]	0.226	13.110	0.348	3.933	0.355	5.976	0.434	5.444	0.297	6.574
IDR [80]	0.231	11.100	0.324	3.866	0.363	5.850	0.453	4.634	0.300	6.683
IRSDE [40]	0.230	11.391	0.351	3.809	0.357	5.874	0.427	4.134	0.285	6.289
AutoDIR [20]	0.231	10.800	0.366	3.910	**0.373**	5.831	**0.470**	9.881	0.308	6.493
DA-CLIP [41]	0.232	10.604	0.334	3.720	0.361	5.864	0.460	6.531	0.310	**6.058**
GOUB [73]	0.231	11.566	**0.373**	3.928	0.360	5.853	0.458	**4.104**	0.278	6.303
ConvIR [11]	**0.236**	**10.280**	0.370	3.723	0.364	**5.813**	0.446	4.645	**0.313**	6.465
DeepSNNet [12]	0.231	11.446	0.348	3.896	0.367	5.882	0.436	4.662	0.301	6.525
AWRaCLe [52]	0.232	12.016	0.366	3.796	0.363	5.898	0.426	4.649	0.306	6.516
MaIR [30]	0.234	10.804	0.350	**3.666**	0.363	5.890	0.245	22.446	0.284	6.590
RDBM	**0.238**	**9.559**	**0.397**	**3.663**	**0.396**	**5.482**	**0.483**	**3.973**	**0.343**	**5.671**

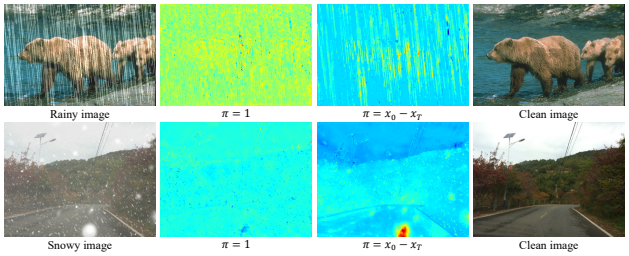


Figure 8. Visualization of noise maps on different $\boldsymbol{\pi}$.

5.6. Noise Maps Visualization

To further elucidate the superiority of our method, we visualize the predicted noise maps generated at a random time point in the reverse process of bridge models under different settings of $\boldsymbol{\pi}$, as depicted in Fig. 8. Obviously, naive diffusion bridge ($\boldsymbol{\pi} = 1$) blindly conducts global noise removal for the reconstruction of missing details. In contrast, our method ($\boldsymbol{\pi} = \mathbf{x}_0 - \mathbf{x}_T$) performs adaptive restoration, as the noise maps are concentrated in degraded regions while remaining relatively smooth in non-degraded areas. In summary, our method can adaptively restore degradation in different regions, showcasing its high flexibility.

5.7. Image Translation and Inpainting

RDBM owns distinct advantages in mapping the data distribution to the prior distribution, thereby enabling extensive validation on similar computer vision tasks. To this end, we expand our experimental settings on image-to-image translation and image inpainting to fully demonstrate the potential of our method. The former aims to transform an input image from one domain to another while preserving certain essential semantic or structural features. The latter focuses on filling in missing or corrupted regions within an image. Specifically, we adopt the widely used edge2handlebags [17] dataset for image-to-image translation and apply the Celebrate-HQ dataset [21] with masks provided in [33] for image inpainting. All these datasets are scaled to 256 × 256. We additionally employ Fréchet Inception Distance (FID) [14] for evaluation. Qualitative comparisons and quantitative results are presented in Fig. 7 and Tab. 9, respectively. Clearly, our method achieves the best visual effects and the best metrics assessments.

Table 9. Quantitative results of image translation and inpainting.

Method	Image Translation [17]				Image Inpainting [21]			
	Edges→Handbags-256×256				Celebrate-HQ-256×256			
	PSNR↑	SSIM↑	FID↓	LPIPS↓	PSNR↑	SSIM↑	FID↓	LPIPS↓
DDPM [62]	8.39	0.447	8.39	0.412	19.22	0.746	0.526	0.126
ReFlow [37]	10.46	0.442	5.76	0.374	23.53	0.822	0.307	0.149
BBDM [29]	14.75	0.635	7.46	0.248	20.36	0.653	0.386	0.169
I^2SB [34]	12.57	0.615	6.64	0.357	27.34	0.890	0.379	**0.054**
RDDM [36]	14.66	0.645	**5.72**	**0.256**	23.94	0.852	**0.167**	0.119
GOUB [73]	**16.58**	**0.700**	8.76	0.288	**31.56**	**0.920**	0.321	0.065
RDBM	**19.26**	**0.738**	**5.38**	**0.224**	**37.88**	**0.965**	**0.147**	**0.031**

6. Conclusion

In this paper, we propose Residual Diffusion Bridge Model, termed as RDBM. Specifically, we theoretically reformulate the stochastic differential equations of generalized diffusion bridge and derive the analytical formulas of its forward and reverse processes. Crucially, we leverage the residual from given distributions to modulate the noise injection and removal, enabling adaptive restoration of degraded regions while preserving intact others. Furthermore, we unravel the fundamental mathematical essence of existing bridge models, all of which are special cases of RDBM, and empirically demonstrate the superiority of our proposed models. Extensive experiments are conducted to demonstrate the state-of-the-art performance of our models both qualitatively and quantitatively across diverse image restoration tasks.

References

[1] Michael S Albergo, Nicholas M Boffi, and Eric Vanden-Eijnden. Stochastic interpolants: A unifying framework for flows and diffusions. *arXiv preprint arXiv:2303.08797*, 2023. 21

[2] Codruta O Ancuti, Cosmin Ancuti, Mateu Sbert, and Radu Timofte. Dense-haze: A benchmark for image dehazing with dense-haze and haze-free images. In *ICIP*, pages 1014–1018. IEEE, 2019. 5, 23

[3] Codruta O. Ancuti, Cosmin Ancuti, and Radu Timofte. NH-HAZE: an image dehazing benchmark with non-homogeneous hazy and haze-free images. In *CVPR Workshops*, 2020. 5, 23

[4] Yunhao Ba, Howard Zhang, Ethan Yang, Akira Suzuki, Arnold Pfahnl, Chethan Chinder Chandrappa, Celso de Melo, Suya You, Stefano Soatto, Alex Wong, and Achuta Kadambi. Not just streaks: Towards ground truth for single image deraining. In *ECCV*, 2022. 5, 23

[5] Arpit Bansal, Eitan Borgnia, Hong-Min Chu, Jie Li, Hamid Kazemi, Furong Huang, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Cold diffusion: Inverting arbitrary image transforms without noise. *NeurIPS*, 36:41259–41282, 2023. 2

[6] Wei-Ting Chen, Hao-Yu Fang, Cheng-Lin Hsieh, Cheng-Che Tsai, I Chen, Jian-Jiun Ding, Sy-Yen Kuo, et al. All snow removed: Single image desnowing algorithm using hierarchical dual-tree complex wavelet representation and contradict channel loss. In *ICCV*, pages 4196–4205, 2021. 5, 23

[7] Xiang Chen, Jinshan Pan, Jiangxin Dong, and Jinhui Tang. Towards unified deep image deraining: A survey and a new benchmark. *IEEE TPAMI*, 2025. 1

[8] Hyungjin Chung, Jeongsol Kim, Sehui Kim, and Jong Chul Ye. Parallel diffusion models of operator and image for blind inverse problems. In *CVPR*, pages 6059–6069, 2023. 1

[9] Hyungjin Chung, Jeongsol Kim, Michael Thompson Mccann, Marc Louis Klasky, and Jong Chul Ye. Diffusion posterior sampling for general noisy inverse problems. In *ICLR*, 2023. 1

[10] Christophe De Vleeschouwer Cosmin Ancuti, Codruta O. Ancuti. D-hazy: A dataset to evaluate quantitatively dehazing algorithms. In *ICIP*, 2016. 5, 23

[11] Yuning Cui, Wenqi Ren, Xiaochun Cao, and Alois Knoll. Revitalizing convolutional network for image restoration. *IEEE TPAMI*, 46(12):9423–9438, 2024. 5, 6, 7, 8, 28

[12] Xin Deng, Chenxiao Zhang, Lai Jiang, Jingyuan Xia, and Mai Xu. Deepsn-net: Deep semi-smooth newton driven network for blind image restoration. *IEEE TPAMI*, 2025. 5, 6, 7, 8, 28

[13] Bhawna Goyal, Ayush Dogra, Dawa Chyophel Lepcha, Vishal Goyal, Ahmed Alkhayyat, Jasgurpreet Singh Chohan, and Vinay Kukreja. Recent advances in image dehazing: Formal analysis to automated approaches. *Information Fusion*, 104:102151, 2024. 1

[14] Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. Gans trained by a two time-scale update rule converge to a local nash equilibrium. *NeurIPS*, 30, 2017. 8

[15] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *NeurIPS*, 33:6840–6851, 2020. 1, 2

[16] Quan Huynh-Thu and Mohammed Ghanbari. Scope of validity of psnr in image/video quality assessment. *Electron. Lett.*, 44(13):800–801, 2008. 5

[17] Phillip Isola, Jun-Yan Zhu, Tinghui Zhou, and Alexei A Efros. Image-to-image translation with conditional adversarial networks. In *CVPR*, pages 1125–1134, 2017. 8

[18] Junjun Jiang, Zengyuan Zuo, Gang Wu, Kui Jiang, and Xianming Liu. A survey on all-in-one image restoration: Taxonomy, evaluation and future trends. *arXiv preprint arXiv:2410.15067*, 2024. 1

[19] Kui Jiang, Zhongyuan Wang, Peng Yi, Chen Chen, Baojin Huang, Yimin Luo, Jiayi Ma, and Junjun Jiang. Multi-scale progressive fusion network for single image deraining. In *CVPR*, 2020. 5, 23

[20] Yitong Jiang, Zhaoyang Zhang, Tianfan Xue, and Jinwei Gu. Autodir: Automatic all-in-one image restoration with latent diffusion. In *ECCV*, pages 340–359. Springer, 2024. 5, 6, 7, 8, 28

[21] Tero Karras, Timo Aila, Samuli Laine, and Jaakko Lehtinen. Progressive growing of gans for improved quality, stability, and variation. In *ICLR*, 2018. 8

[22] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *NeurIPS*, 35:26565–26577, 2022. 1

[23] Tero Karras, Miika Aittala, Jaakko Lehtinen, Janne Hellsten, Timo Aila, and Samuli Laine. Analyzing and improving the training dynamics of diffusion models. In *CVPR*, pages 24174–24184, 2024. 2

[24] Taewoo Kim, Hoonhee Cho, and Kuk-Jin Yoon. Frequency-aware event-based video deblurring for real-world motion blur. In *CVPR*, pages 24966–24976, 2024. 1

[25] Diederik P Kingma and Max Welling. Auto-encoding variational {Bayes}. In *ICLR*, 2014. 4

[26] Chulwoo Lee, Chul Lee, and Chang-Su Kim. Contrast enhancement based on layered difference representation of 2d histograms. *IEEE TIP*, 22(12):5372–5384, 2013. 5, 23

[27] Boyi Li, Wenqi Ren, Dengpan Fu, Dacheng Tao, Dan Feng, Wenjun Zeng, and Zhangyang Wang. Benchmarking single-image dehazing and beyond. *IEEE TIP*, 28(1):492–505, 2018. 5, 23

[28] Boyun Li, Xiao Liu, Peng Hu, Zhongqin Wu, Jiancheng Lv, and Xi Peng. All-in-one image restoration for unknown corruption. In *CVPR*, pages 17452–17462, 2022. 5, 6, 7, 8, 28

[29] Bo Li, Kaitao Xue, Bin Liu, and Yu-Kun Lai. Bbdm: Image-to-image translation with brownian bridge diffusion models. In *CVPR*, pages 1952–1961, 2023. 2, 4, 8, 21

[30] Boyun Li, Haiyu Zhao, Wenxin Wang, Peng Hu, Yuanbiao Gou, and Xi Peng. Mair: A locality-and continuity-preserving mamba for image restoration. In *CVPR*, pages 7491–7501, 2025. 5, 6, 7, 8, 28

[31] Wei Li, Qiming Zhang, Jing Zhang, Zhen Huang, Xinmei Tian, and Dacheng Tao. Toward real-world single image deraining: A new benchmark and beyond. *arXiv preprint arXiv:2206.05514*, 2022. 5, 23

[32] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow matching for generative modeling. In *ICLR*, 2023. 2, 4, 21

[33] Guilin Liu, Fitsum A Reda, Kevin J Shih, Ting-Chun Wang, Andrew Tao, and Bryan Catanzaro. Image inpainting for irregular holes using partial convolutions. In *ECCV*, pages 85–100, 2018. 8

[34] Guan-Horng Liu, Arash Vahdat, De-An Huang, Evangelos A Theodorou, Weili Nie, and Anima Anandkumar. I2sb: image-to-image schrodinger bridge. In *ICML*, pages 22042–22062, 2023. 2, 4, 8, 21

[35] Jiaying Liu, Xu Dejia, Wenhan Yang, Minhao Fan, and Haofeng Huang. Benchmarking low-light image enhancement and beyond. *International Journal of Computer Vision*, 129:1153–1184, 2021. 5, 23

[36] Jiawei Liu, Qiang Wang, Huijie Fan, Yinong Wang, Yandong Tang, and Liangqiong Qu. Residual denoising diffusion models. In *CVPR*, pages 2773–2783, 2024. 1, 2, 8

[37] Xingchao Liu, Chengyue Gong, et al. Flow straight and fast: Learning to generate and transfer data with rectified flow. In *ICLR*, 2022. 2, 4, 8, 21

[38] Yun-Fu Liu, Da-Wei Jaw, Shih-Chia Huang, and Jenq-Neng Hwang. Desnownet: Context-aware deep network for snow removal. *IEEE TIP*, 27(6):3064–3073, 2018. 5, 23

[39] Weijian Luo, Tianyang Hu, Shifeng Zhang, Jiacheng Sun, Zhenguo Li, and Zhihua Zhang. Diff-instruct: A universal approach for transferring knowledge from pre-trained diffusion models. *NeurIPS*, 36:76525–76546, 2023. 1

[40] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, and Jens Sjölund. Image restoration with mean-reverting stochastic differential equations. In *ICML*, pages 23045–23066, 2023. 1, 2, 4, 5, 6, 7, 8, 28

[41] Ziwei Luo, Fredrik K Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas B Schön. Controlling vision-language models for multi-task image restoration. In *ICLR*, 2024. 5, 6, 7, 8, 28

[42] Ziwei Luo, Fredrik Gustafsson, Zheng Zhao, Jens Sjölund, and Thomas Schön. Taming diffusion models for image restoration: a review. *Philosophical Transactions A*, 383 (2299):20240358, 2025. 1

[43] Jiaqi Ma, Tianheng Cheng, Guoli Wang, Qian Zhang, Xinggang Wang, and Lefei Zhang. Prores: Exploring degradation-aware visual prompt for universal image restoration. *CoRR*, 2023. 5, 6, 7, 8, 28

[44] Kede Ma, Kai Zeng, and Zhou Wang. Perceptual quality assessment for multi-exposure image fusion. *IEEE TIP*, 24 (11):3345–3356, 2015. 5, 23

[45] Zhiyuan Ma, Yuzhu Zhang, Guoli Jia, Liangliang Zhao, Yichao Ma, Mingjie Ma, Gaofeng Liu, Kaiyan Zhang, Ning Ding, Jianjun Li, et al. Efficient diffusion models: A comprehensive survey from principles to practices. *IEEE TPAMI*, 2025. 1

[46] Aboli Marathe, Pushkar Jain, Rahee Walambe, and Ketan Kotecha. Restorex-ai: A contrastive approach towards guiding image restoration via explainable ai systems. In *CVPR*, pages 3030–3039, 2022. 1

[47] Anish Mittal, Rajiv Soundararajan, and Alan C Bovik. Making a "completely blind" image quality analyzer. *IEEE Trans. Signal Process.*, 20(3):209–212, 2012. 7

[48] Seungjun Nah, Tae Hyun Kim, and Kyoung Mu Lee. Deep multi-scale convolutional neural network for dynamic scene deblurring. In *CVPR*, pages 3883–3891, 2017. 5, 23

[49] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019. 5

[50] Vaishnav Potlapalli, Syed Waqas Zamir, Salman H Khan, and Fahad Shahbaz Khan. Promptir: Prompting for all-in-one image restoration. *NeurIPS*, 36:71275–71293, 2023. 5, 6, 7, 8, 28

[51] Rui Qian, Robby T Tan, Wenhan Yang, Jiajun Su, and Jiaying Liu. Attentive generative adversarial network for raindrop removal from a single image. In *CVPR*, pages 2482–2491, 2018. 5, 23

[52] Sudarshan Rajagopalan and Vishal M Patel. Awracle: All-weather image restoration using visual in-context learning. In *AAAI*, pages 6675–6683, 2025. 5, 6, 7, 8, 28

[53] Jaesung Rim, Haeyun Lee, Jucheol Won, and Sunghyun Cho. Real-world blur dataset for learning and benchmarking deblurring algorithms. In *ECCV*, 2020. 5, 23

[54] Hannes Risken. Fokker-planck equation. In *The Fokker-Planck equation: methods of solution and applications*, pages 63–95. Springer, 1989. 13

[55] L Chris G Rogers and David Williams. *Diffusions, Markov processes, and martingales*. Cambridge university press, 2000. 2

[56] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *MICCAI*, pages 234–241. Springer, 2015. 5

[57] Simo Särkkä and Arno Solin. *Applied stochastic differential equations*. Cambridge University Press, 2019. 2

[58] Zhenning Shi, Chen Xu, Changsheng Dong, Bin Pan, Along He, Tao Li, Huazhu Fu, et al. Resfusion: Denoising diffusion probabilistic models for image restoration based on prior residual noise. *NeurIPS*, 37:130664–130693, 2024. 2

[59] Jake Snell, Karl Ridgeway, Renjie Liao, Brett D Roads, Michael C Mozer, and Richard S Zemel. Learning to generate images with perceptual similarity metrics. In *ICIP*, pages 4277–4281. IEEE, 2017. 5

[60] Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *ICML*, pages 2256–2265. pmlr, 2015. 2

[61] Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *ICLR*, 2021. 1, 2

[62] Yang Song, Jascha Sohl-Dickstein, Diederik P Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2020. 2, 8, 20

[63] Chunwei Tian, Menghua Zheng, Wangmeng Zuo, Shichao Zhang, Yanning Zhang, and Chia-Wen Lin. A cross transformer for image denoising. *Information Fusion*, 102: 102043, 2024. 1

[64] Hebaixu Wang, Jing Zhang, Haonan Guo, Di Wang, Jiayi Ma, and Bo Du. Dgsolver: Diffusion generalist solver with universal posterior sampling for image restoration. *arXiv preprint arXiv:2504.21487*, 2025. 1

[65] Liyan Wang, Weixiang Zhou, Cong Wang, Kin-Man Lam, Zhixun Su, and Jinshan Pan. Deep learning-driven ultra-high-definition image restoration: A survey. *arXiv preprint arXiv:2505.16161*, 2025. 1

[66] Shuhang Wang, Jin Zheng, Hai-Miao Hu, and Bo Li. Naturalness preserved enhancement algorithm for non-uniform illumination images. *IEEE TIP*, 22(9):3538–3548, 2013. 5, 23

[67] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. Image quality assessment: from error visibility to structural similarity. *IEEE TIP*, 13(4):600–612, 2004. 5

[68] Chen Wei, Wenjing Wang, Wenhan Yang, and Jiaying Liu. Deep retinex decomposition for low-light enhancement. *arXiv preprint arXiv:1808.04560*, 2018. 5, 23

[69] Jay Whang, Mauricio Delbracio, Hossein Talebi, Chitwan Saharia, Alexandros G Dimakis, and Peyman Milanfar. Deblurring via stochastic refinement. In *CVPR*, pages 16293–16303, 2022. 2

[70] Rongyuan Wu, Tao Yang, Lingchen Sun, Zhengqiang Zhang, Shuai Li, and Lei Zhang. Seesr: Towards semantics-aware real-world image super-resolution. In *CVPR*, pages 25456–25467, 2024. 1

[71] Bin Xia, Yulun Zhang, Shiyin Wang, Yitong Wang, Xinglong Wu, Yapeng Tian, Wenming Yang, and Luc Van Gool. Diffir: Efficient diffusion model for image restoration. In *ICCV*, pages 13095–13105, 2023. 1, 2

[72] Wenhan Yang, Robby T Tan, Jiashi Feng, Jiaying Liu, Zongming Guo, and Shuicheng Yan. Deep joint rain detection and removal from a single image. In *CVPR*, pages 1357–1366, 2017. 5, 23

[73] Conghan Yue, Zhengwei Peng, Junlong Ma, Shiyan Du, Pengxu Wei, and Dongyu Zhang. Image restoration through generalized ornstein-uhlenbeck bridge. In *ICML*, pages 58068–58089, 2024. 2, 4, 5, 6, 7, 8, 21, 28

[74] Zongsheng Yue, Jianyi Wang, and Chen Change Loy. Efficient diffusion model for image restoration by residual shifting. *IEEE TPAMI*, 2024. 2

[75] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, Ming-Hsuan Yang, and Ling Shao. Learning enriched features for real image restoration and enhancement. In *ECCV*, pages 492–511. Springer, 2020. 2

[76] Syed Waqas Zamir, Aditya Arora, Salman Khan, Munawar Hayat, Fahad Shahbaz Khan, and Ming-Hsuan Yang. Restormer: Efficient transformer for high-resolution image restoration. In *CVPR*, pages 5728–5739, 2022. 5, 6, 7, 8, 28

[77] Bingliang Zhang, Wenda Chu, Julius Berner, Chenlin Meng, Anima Anandkumar, and Yang Song. Improving diffusion inverse problem solving with decoupled noise annealing. In *CVPR*, pages 20895–20905, 2025. 1

[78] He Zhang and Vishal M Patel. Density-aware single image de-raining using a multi-stream dense network. In *CVPR*, pages 695–704, 2018. 5, 23

[79] Jing Zhang, Yang Cao, Shuai Fang, Yu Kang, and Chang Wen Chen. Fast haze removal for nighttime image using maximum reflectance prior. In *CVPR*, pages 7418–7426, 2017. 5, 23

[80] Jinghao Zhang, Jie Huang, Mingde Yao, Zizheng Yang, Hu Yu, Man Zhou, and Feng Zhao. Ingredient-oriented multi-degradation learning for image restoration. In *CVPR*, pages 5825–5835, 2023. 5, 6, 7, 8, 28

[81] Zhilu Zhang, Shuohao Zhang, Renlong Wu, Wangmeng Zuo, Radu Timofte, Xiaoxia Xing, Hyunhee Park, Sejun Song, Changho Kim, Xiangyu Kong, et al. Ntire 2024 challenge on bracketing image restoration and enhancement: Datasets methods and results. In *CVPR*, pages 6153–6166, 2024. 1

[82] Dian Zheng, Xiao-Ming Wu, Shuzhou Yang, Jian Zhang, Jian-Fang Hu, and Wei-Shi Zheng. Selective hourglass mapping for universal image restoration based on diffusion model. In *CVPR*, pages 25445–25455, 2024. 2

[83] Linqi Zhou, Aaron Lou, Samar Khanna, and Stefano Ermon. Denoising diffusion bridge models. In *ICLR*, 2024. 2, 4, 20

[84] Yuqian Zhou, David Ren, Neil Emerton, Sehoon Lim, and Timothy Large. Image restoration for under-display camera. In *CVPR*, pages 9179–9188, 2021. 7, 25

[85] Hancheng Zhu, Leida Li, Jinjian Wu, Weisheng Dong, and Guangming Shi. Metaiqa: Deep meta-learning for no-reference image quality assessment. In *CVPR*, pages 14143–14152, 2020. 7

[86] Kaizhen Zhu, Mokai Pan, Zhechuan Yu, Jingya Wang, Jingyi Yu, and Ye Shi. Diffusion bridge or flow matching? a unifying framework and comparative analysis. *arXiv preprint arXiv:2509.24531*, 2025. 2

[87] Yuanzhi Zhu, Kai Zhang, Jingyun Liang, Jiezhang Cao, Bihan Wen, Radu Timofte, and Luc Van Gool. Denoising diffusion models for plug-and-play image restoration. In *CVPR*, pages 1219–1229, 2023. 1

Residual Diffusion Bridge Model for Image Restoration

Supplementary Material

Appendix Contents

A. Doob's h transform

Theorem 1 *For a given SDE:*

$$\mathrm{d}\mathbf{x}_t = \mathbf{f}\left(\mathbf{x}_t, t\right)\mathrm{d}t + g_t \mathrm{d}\mathbf{w}_t, \qquad \mathbf{x}_0 \sim p\left(\mathbf{x}_0\right), \tag{A.1}$$

For a fixed $\mathbf{x}_T$, the evolution of conditional probability $p(\mathbf{x}_t \mid \mathbf{x}_T)$ follows:

$$\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) + g_t^2 \mathbf{h}(\mathbf{x}_t, t, \mathbf{x}_T, T)\right]\mathrm{d}t + g_t \mathrm{d}\mathbf{w}_t, \qquad \mathbf{x}_0 \sim p\left(\mathbf{x}_0 \mid \mathbf{x}_T\right), \tag{A.2}$$

where $\mathbf{h}(\mathbf{x}_t, t, \mathbf{x}_T, T) = \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_T \mid \mathbf{x}_t)$.

Proof: In theory, $p(\mathbf{x}_t \mid \mathbf{x}_0)$ and $p(\mathbf{x}_T \mid \mathbf{x}_t)$ satisfy the Kolmogorov Forward Equation (KFE) and Kolmogorov Backward Equation (KBE), respectively [54], as formulated below:

$$\frac{\partial}{\partial t} p(\mathbf{x}_t \mid \mathbf{x}_0) = -\nabla_{\mathbf{x}_t} \cdot [\mathbf{f}(\mathbf{x}_t, t) p(\mathbf{x}_t \mid \mathbf{x}_0)] + \frac{1}{2} g_t^2 \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{x}_0), \tag{A.3}$$

$$-\frac{\partial}{\partial t} p(\mathbf{x}_T \mid \mathbf{x}_t) = \mathbf{f}(\mathbf{x}_t, t) \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_T \mid \mathbf{x}_t) + \frac{1}{2} g_t^2 \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_T \mid \mathbf{x}_t). \tag{A.4}$$

Using Bayes' rule, we have:

$$\begin{aligned} p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) &= \frac{p(\mathbf{x}_T \mid \mathbf{x}_t, \mathbf{x}_0) p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \\ &= \frac{p(\mathbf{x}_T \mid \mathbf{x}_t) p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \end{aligned} \tag{A.5}$$

Therefore, the derivative of conditional transition probability $p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)$ with time follows:

$$\begin{aligned} \frac{\partial}{\partial t} p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) &= \frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \frac{\partial}{\partial t} p(\mathbf{x}_T \mid \mathbf{x}_t) + \frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \frac{\partial}{\partial t} p(\mathbf{x}_t \mid \mathbf{x}_0) \\ &= \frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \left[-\mathbf{f}(\mathbf{x}_t, t) \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_T \mid \mathbf{x}_t) - \frac{1}{2} g_t^2 \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_T \mid \mathbf{x}_t)\right] \\ &\quad + \frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \left\{-\nabla_{\mathbf{x}_t} \cdot [\mathbf{f}(\mathbf{x}_t, t) p(\mathbf{x}_t \mid \mathbf{x}_0)] + \frac{1}{2} g_t^2 \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{x}_0)\right\} \\ &= -\left[\frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \mathbf{f}(\mathbf{x}_t, t) \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_T \mid \mathbf{x}_t) + \frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \mathbf{f}(\mathbf{x}_t, t) \nabla_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{x}_0)\right. \\ &\quad \left. + \frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} p(\mathbf{x}_t \mid \mathbf{x}_0) \nabla_{\mathbf{x}_t} \cdot \mathbf{f}(\mathbf{x}_t, t)\right] \\ &\quad + \frac{1}{2} g_t^2 \left[\frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{x}_0) - \frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_T \mid \mathbf{x}_t)\right] \\ &= -\left[\mathbf{f}(\mathbf{x}_t, t) \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) + p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) \cdot \nabla_{\mathbf{x}_t} \mathbf{f}(\mathbf{x}_t, t)\right] \\ &\quad + \frac{1}{2} g_t^2 \left[\frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{x}_0) - \frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_T \mid \mathbf{x}_t)\right] \\ &= -\nabla_{\mathbf{x}_t} \cdot [\mathbf{f}(\mathbf{x}_t, t) p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)] \\ &\quad + \frac{1}{2} g_t^2 \left[\frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_t \mid \mathbf{x}_0) - \frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)} \nabla_{\mathbf{x}_t} \cdot \nabla_{\mathbf{x}_t} p(\mathbf{x}_T \mid \mathbf{x}_t)\right] \end{aligned} \tag{A.6}$$

For the second term, we have:

$$
\begin{aligned}
&\frac{1}{2}g_t^2\left[\frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}\cdot\nabla_{\mathbf{x}_t}p(\mathbf{x}_t \mid \mathbf{x}_0) - \frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}\cdot\nabla_{\mathbf{x}_t}p(\mathbf{x}_T \mid \mathbf{x}_t)\right] \\
=&\frac{1}{2}g_t^2\left[\frac{p(\mathbf{x}_T \mid \mathbf{x}_t)}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}\cdot\nabla_{\mathbf{x}_t}p(\mathbf{x}_t \mid \mathbf{x}_0) + \frac{1}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}p(\mathbf{x}_T \mid \mathbf{x}_t)\cdot\nabla_{\mathbf{x}_t}\, p(\mathbf{x}_t \mid \mathbf{x}_0)\right. \\
&\left.+\frac{1}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}p(\mathbf{x}_T \mid \mathbf{x}_t)\cdot\nabla_{\mathbf{x}_t}\, p(\mathbf{x}_t \mid \mathbf{x}_0) + \frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}\cdot\nabla_{\mathbf{x}_t}p(\mathbf{x}_T \mid \mathbf{x}_t)\right] \\
&- g_t^2\left[\frac{1}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}p(\mathbf{x}_T \mid \mathbf{x}_t)\cdot\nabla_{\mathbf{x}_t}\, p(\mathbf{x}_t \mid \mathbf{x}_0) + \frac{p(\mathbf{x}_t \mid \mathbf{x}_0)}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}\cdot\nabla_{\mathbf{x}_t}p(\mathbf{x}_T \mid \mathbf{x}_t)\right] \\
=&\frac{1}{2}g_t^2\left[\frac{1}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_T \mid \mathbf{x}_t)\nabla_{\mathbf{x}_t}p(\mathbf{x}_t \mid \mathbf{x}_0)] + \frac{1}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0)\nabla_{\mathbf{x}_t}p(\mathbf{x}_T \mid \mathbf{x}_t)]\right] \\
&- g_t^2\frac{1}{p(\mathbf{x}_T \mid \mathbf{x}_0)}\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0)\nabla_{\mathbf{x}_t}p(\mathbf{x}_T \mid \mathbf{x}_t)] \\
=&\frac{1}{2}g_t^2\left[\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t \mid \mathbf{x}_0)] + \nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_T \mid \mathbf{x}_t)]\right] \\
&- g_t^2\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_T \mid \mathbf{x}_t)] \\
=&\frac{1}{2}g_t^2\left[\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)]\right] - g_t^2\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_T \mid \mathbf{x}_t)] \\
=&\frac{1}{2}g_t^2\nabla_{\mathbf{x}_t}\cdot\nabla_{\mathbf{x}_t}p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) - g_t^2\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_T \mid \mathbf{x}_t)]
\end{aligned}
\tag{A.7}
$$

Bring it back to (A.6):

$$
\begin{aligned}
\frac{\partial}{\partial t}p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) &= -\nabla_{\mathbf{x}_t}\cdot[\mathbf{f}(\mathbf{x}_t, t)p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)] + \frac{1}{2}g_t^2\nabla_{\mathbf{x}_t}\cdot\nabla_{\mathbf{x}_t}p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T) \\
&\quad - g_t^2\nabla_{\mathbf{x}_t}\cdot[p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_T \mid \mathbf{x}_t)] \\
&= -\nabla_{\mathbf{x}_t}\cdot\left[[\mathbf{f}(\mathbf{x}_t, t) + g_t^2\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_T \mid \mathbf{x}_t)]p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)\right] + \frac{1}{2}g_t^2\nabla_{\mathbf{x}_t}\cdot\nabla_{\mathbf{x}_t}p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)
\end{aligned}
\tag{A.8}
$$

This is the definition of FP equation of conditional transition probability $p(\mathbf{x}_t \mid \mathbf{x}_0, \mathbf{x}_T)$, which represents the evolution that follows the SDE:

$$
\mathrm{d}\mathbf{x}_t = \left[\mathbf{f}(\mathbf{x}_t, t) + g_t^2\nabla_{\mathbf{x}_t}\log p(\mathbf{x}_T \mid \mathbf{x}_t)\right]\mathrm{d}t + g_t\mathrm{d}\mathbf{w}_t \tag{A.9}
$$

This concludes the proof of the **Theorem 1** in Sec. 3.1.

B. Mean-Reverting Ornstein–Uhlenbeck Process

Theorem 2 *The SDE formulation of the Ornstein–Uhlenbeck process with its predefined coefficients θ_t, σ_t is:*

$$
d\mathbf{x}_t = \theta_t(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sigma_t dw_t, \tag{B.1}
$$

where $\boldsymbol{\mu}$ represents the mean value that $\mathbf{x}_t$ will approximate at $t = T$. The solution of OU process can be calculated as:

$$
\mathbf{x}_t = \boldsymbol{\mu} + (x_0 - \boldsymbol{\mu})e^{-\int_0^t \theta_s ds} + e^{-\int_0^t \theta_s ds}\int_0^t \sigma_s e^{\int_0^s \theta_u du}dw_s, \tag{B.2}
$$

Proof. We define a surrogate differentiable function $\psi(\mathbf{x}, t) = \mathbf{x}e^{\int_0^t \theta_z dz} = \mathbf{x}e^{\overline{\theta}_t}$ and expand it by $Itô$ formula:

$$
\begin{aligned}
d\psi(\mathbf{x}, t) &= \frac{\partial\psi}{\partial t}(\mathbf{x}, t)dt + \frac{\partial\psi}{\partial\mathbf{x}}(\mathbf{x}, t)d\mathbf{x} + \frac{1}{2}\frac{\partial^2\psi}{\partial\mathbf{x}^2}(\mathbf{x}, t)d\mathbf{x}^2 \\
&= \theta_t\mathbf{x}e^{\overline{\theta}_t}dt + e^{\overline{\theta}_t}(\theta_t(\boldsymbol{\mu} - \mathbf{x})dt + \sigma_t dw_t) \\
&= \boldsymbol{\mu}\theta_t e^{\overline{\theta}_t}dt + \sigma_t e^{\overline{\theta}_t}dw_t
\end{aligned}
\tag{B.3}
$$

Then, we can solve $\mathbf{x}_t$ conditioned on $\mathbf{x}_s$ where $s < t$, as:

$$\psi(\mathbf{x}_t, t) - \psi(\mathbf{x}_s, s) = \int_s^t \boldsymbol{\mu}\theta_z e^{\overline{\theta}_z} dz + \int_s^t \sigma_z e^{\overline{\theta}_z} dw_z, \tag{B.4}$$

$$\mathbf{x}_t e^{\overline{\theta}_t} - \mathbf{x}_s e^{\overline{\theta}_s} = \boldsymbol{\mu}(e^{\overline{\theta}_t} - e^{\overline{\theta}_s}) + \int_s^t \sigma_z e^{\overline{\theta}_z} dw_z, \tag{B.5}$$

$$\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_s - \boldsymbol{\mu}) e^{-\theta_{s:t}} + \int_s^t \sigma_z e^{-\overline{\theta}_{z:t}} dw_z. \tag{B.6}$$

where $-\theta_{s:t} = -\int_s^t \theta_z dz$, and thus we complete the proof. The expectation and variance of Eq. (B.6) can be rewritten:

$$E[x_t] = \boldsymbol{\mu} + (x_s - \boldsymbol{\mu}) e^{-\theta_{s:t}}, \tag{B.7}$$

$$Var[x_t] = \int_s^t \sigma_z^2 e^{-2\overline{\theta}_{z:t}} dz, \tag{B.8}$$

This concludes the derivations in Sec. 3.2.

C. RDBM Formulation

Proposition 1: Let $\mathbf{x}_t$ be a finite random variable governed by the generalized OU process, with terminal condition $\mathbf{x}_T = \boldsymbol{\mu}$. The evolution of its marginal distribution $p(\mathbf{x}_t \mid \mathbf{x}_T)$ satisfies the following SDE under a fixed drift-to-diffusion coefficient ratio λ:

$$d\mathbf{x}_t = \theta_t \coth(\theta_{t:T})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sqrt{2\boldsymbol{\pi}^2 \lambda \theta_t} dw_t, \tag{C.1}$$

where $\theta_{t:T} = \int_t^T \theta_z dz$ and $\boldsymbol{\pi} \in \mathbb{R}$ is the predefined parameter.
Proof: First, we define a generalized OU process with the properties of mean-reverting:

$$d\mathbf{x}_t = \theta_t(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t dw_t. \tag{C.2}$$

In our formulation, $\boldsymbol{\pi} = \boldsymbol{\mu} - \mathbf{x}_0$ is considered as the residual of given distributions. When $\boldsymbol{\pi} = 1$ or $\boldsymbol{\pi} = 0$, Eq. (C.2) can degenerate to other bridge models, as discussed in Suppl. F. Here, we solve this SDE step by step, akin to Suppl. B. First, we define a surrogate differentiable function $\psi(\mathbf{x}, t) = \mathbf{x} e^{\int_0^t \theta_z dz} = \mathbf{x} e^{\overline{\theta}_t}$ and expand it by $Itô$ formula:

$$\begin{aligned} d\psi(\mathbf{x}, t) &= \frac{\partial \psi}{\partial t}(\mathbf{x}, t)dt + \frac{\partial \psi}{\partial \mathbf{x}}(\mathbf{x}, t)d\mathbf{x} + \frac{1}{2}\frac{\partial^2 \psi}{\partial \mathbf{x}^2}(\mathbf{x}, t)d\mathbf{x}^2 \\ &= \theta_t \mathbf{x} e^{\overline{\theta}_t} dt + e^{\overline{\theta}_t}(\theta_t(\boldsymbol{\mu} - \mathbf{x})dt + \boldsymbol{\pi}\sigma_t dw_t) \\ &= \boldsymbol{\mu}\theta_t e^{\overline{\theta}_t} dt + \boldsymbol{\pi}\sigma_t e^{\overline{\theta}_t} dw_t \end{aligned} \tag{C.3}$$

Then, we can solve $\mathbf{x}_t$ conditioned on $\mathbf{x}_s$ where $s < t$, as:

$$\psi(\mathbf{x}_t, t) - \psi(\mathbf{x}_s, s) = \int_s^t \boldsymbol{\mu}\theta_z e^{\overline{\theta}_z} dz + \int_s^t \boldsymbol{\pi}\sigma_z e^{\overline{\theta}_z} dw_z, \tag{C.4}$$

$$\mathbf{x}_t e^{\overline{\theta}_t} - \mathbf{x}_s e^{\overline{\theta}_s} = \boldsymbol{\mu}(e^{\overline{\theta}_t} - e^{\overline{\theta}_s}) + \int_s^t \boldsymbol{\pi}\sigma_z e^{\overline{\theta}_z} dw_z, \tag{C.5}$$

$$\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_s - \boldsymbol{\mu}) e^{-\theta_{s:t}} + \int_s^t \boldsymbol{\pi}\sigma_z e^{-\overline{\theta}_{z:t}} dw_z. \tag{C.6}$$

where $-\theta_{s:t} = -\int_s^t \theta_z dz$. The expectation and variance of Eq. (C.6) can be written as below:

$$E[\mathbf{x}_t] = \boldsymbol{\mu} + (\mathbf{x}_s - \boldsymbol{\mu}) e^{-\theta_{s:t}}, \tag{C.7}$$

$$Var[\mathbf{x}_t] = \boldsymbol{\pi}^2 \int_s^t \sigma_z^2 e^{-2\overline{\theta}_{z:t}} dz, \tag{C.8}$$

To derive the analytical form of Eq. (C.8), we assume that $\lambda = \frac{\sigma_t^2}{2\theta_t}$ is pre-defined stationary variance, and obtain:

$$Var[\mathbf{x}_t] = \lambda \boldsymbol{\pi}^2 \int_s^t 2\theta_z e^{-2\overline{\theta}_{z:t}} dz = \lambda \boldsymbol{\pi}^2 (1 - e^{-2\theta_{s:t}}), \tag{C.9}$$

We can conclude that:

$$p(\mathbf{x}_t|\mathbf{x}_s) \sim \mathcal{N}(\boldsymbol{\mu} + (\mathbf{x}_s - \boldsymbol{\mu}) e^{-\theta_{s:t}}, \lambda \boldsymbol{\pi}^2 (1 - e^{-2\theta_{s:t}})), \tag{C.10}$$

To ensure that the final state of time point $t = T$ conforms to the distribution of low-quality image $\mathbf{x}_T = \boldsymbol{\mu} \sim p_{LQ}(\boldsymbol{x})$, we leverage the Doob's h transform by modifying the forward SDE from Eq. (C.11) to Eq. (C.12):

$$d\mathbf{x}_t = \mathbf{f}(\mathbf{x}, t)dt + g(t)dw_t, \tag{C.11}$$

$$d\mathbf{x}_t = [\mathbf{f}(\mathbf{x}, t) + g(t)^2 \nabla_{\mathbf{x}_t} \log p(\mathbf{x}_T|\mathbf{x}_t)]dt + g(t)dw_t, \tag{C.12}$$

where term $\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_T|x_t)$ can be calculated by setting $s = 0, t = T$ in Eq. (C.10):

$$\nabla_{\mathbf{x}_t} \log p(\mathbf{x}_T|\mathbf{x}_t) = (\boldsymbol{\mu} - \mathbf{x}_t) \frac{e^{-2\theta_{t:T}}}{\lambda \boldsymbol{\pi}^2 (1 - e^{-2\theta_{t:T}})}. \tag{C.13}$$

The mean-reverting OU process turns into a mean-arriving process, which can be formulated as:

$$\begin{aligned}
d\mathbf{x}_t &= (\theta_t + \frac{\sigma_t^2 e^{-2\theta_{t:T}}}{\lambda(1 - e^{-2\theta_{t:T}})})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t dw_t, \\
&= \theta_t (1 + \frac{2e^{-2\theta_{t:T}}}{1 - e^{-2\theta_{t:T}}})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t dw_t \\
&= \theta_t (\frac{1 + e^{-2\theta_{t:T}}}{1 - e^{-2\theta_{t:T}}})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t dw_t, \\
&= \theta_t \coth(\theta_{t:T})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t dw_t, (\sigma_t^2 = 2\lambda\theta_t), \\
&= \theta_t \coth(\theta_{t:T})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sqrt{2\boldsymbol{\pi}^2 \lambda \theta_t} dw_t,
\end{aligned} \tag{C.14}$$

Eq. (C.14) can be converted into an analytical formula as follows. First, we substitute $y_t = \mathbf{x}_t - \boldsymbol{\mu}$, then the SDE of y_t becomes:

$$dy_t = -\theta_t \coth(\theta_{t:T}) y_t dt + \sqrt{2\boldsymbol{\pi}^2 \lambda \theta_t} dw_t, \tag{C.15}$$

Second, we introduce $\Psi_t = \exp(\int_0^t \theta_s \coth(\theta_{s:T})ds)$ as the integrating factor and expand $\Psi_t y_t$ by Itô formula:

$$d(\Psi_t y_t) = \Psi_t dy_t + y_t d\Psi_t + d\Psi_t dy_t, \tag{C.16}$$

Since Ψ is a deterministic function, it satisfies $d\Psi = \Psi \theta_t \coth(\theta_{t:T})$. $d\Psi dy_t$ produces $(dt)^2, dtdw_t$, which are the higher order infinitesimal of dt and can be omitted. Thus, we obtain:

$$d(\Psi_t y_t) = \Psi_t(-\theta_t \coth(\theta_{t:T}) y_t dt + \sqrt{2\boldsymbol{\pi}^2 \lambda \theta_t} dw_t) + \Psi_t \theta_t \coth(\theta_{t:T}) y_t dt = \Psi_t \sqrt{2\boldsymbol{\pi}^2 \lambda \theta_t} dw_t, \tag{C.17}$$

Furthermore, we integrate both sides of Eq. (C.17):

$$\Psi_t y_t = y_0 + \int_0^t \Psi_s \sqrt{2\boldsymbol{\pi}^2 \lambda \theta_s} dw_s. \tag{C.18}$$

Consequently, we have:

$$\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu}) \Psi_t^{e^{-\int_0^t \theta_s \coth(\theta_{s:T})ds}} + \int_0^t \sqrt{2\boldsymbol{\pi}^2 \lambda \theta_s} e^{-\int_s^t \theta_z \coth(\theta_{z:T})dz} dw_s. \tag{C.19}$$

We next analyze the analytical formulation of Ψ_t. Considering the internal integral $\int_0^t \theta_s \coth(\theta_{s:T})ds$ at first, we set $u = \theta_{s:T}$ satisfying $du = -\theta_s ds$:

$$\int_0^t \theta_s \coth(\theta_{s:T})ds = -\int_{\theta_{0:T}}^{\theta_{t:T}} \coth(u)du = -\ln|\sinh(u)|\Big|_{\theta_{0:T}}^{\theta_{t:T}} = \ln|\frac{\sinh(\theta_{0:T})}{\sinh(\theta_{t:T})}|, \tag{C.20}$$

Therefore, the analytical expression of Ψ_t is:

$$\Psi_t = \frac{\sinh(\theta_{0:T})}{\sinh(\theta_{t:T})}. \tag{C.21}$$

Finally, we can compute the closed-form of $\mathbf{x}_t$ in Eq. (C.19):

$$\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\frac{\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})} + \int_0^t \sqrt{2\boldsymbol{\pi}^2\lambda\theta_s}\frac{\sinh(\theta_{t:T})}{\sinh(\theta_{s:T})}dw_s. \tag{C.22}$$

Eq. (C.22) preserves the properties of diffusion bridge models, whose initial state $\mathbf{x}_0$ and final state $\mathbf{x}_T$ are determined. The formulation of variance can be further simplified as follows:

$$\begin{aligned} Var[x_t] = \int_0^t 2\boldsymbol{\pi}^2\lambda\theta_s(\frac{\sinh(\theta_{t:T})}{\sinh(\theta_{s:T})})^2 ds &= 2\boldsymbol{\pi}^2\lambda\sinh^2(\theta_{t:T})\int_{\theta_{0:T}}^{\theta_{t:T}} -\frac{du}{\sinh^2(u)} \\ &= 2\boldsymbol{\pi}^2\lambda\sinh^2(\theta_{t:T})\coth(u)\Big|_{\theta_{0:T}}^{\theta_{t:T}} \\ &= 2\boldsymbol{\pi}^2\lambda\sinh^2(\theta_{t:T})(\coth(\theta_{t:T}) - \coth(\theta_{0:T})) \\ &= 2\boldsymbol{\pi}^2\lambda\sinh^2(\theta_{t:T})(\frac{\sinh(\theta_{0:T}-\theta_{t:T})}{\sinh(\theta_{0:T})\sinh(\theta_{t:T})}) \\ &= 2\boldsymbol{\pi}^2\lambda\frac{\sinh(\theta_{0:t})\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})} \end{aligned} \tag{C.23}$$

The expectation and variance of Eq. (C.22) are:

$$E[\mathbf{x}_t] = \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\frac{\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})}, \tag{C.24}$$

$$Var[\mathbf{x}_t] = 2\boldsymbol{\pi}^2\lambda\frac{\sinh(\theta_{0:t})\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})}. \tag{C.25}$$

This concludes the derivations in Sec. 4.1.

D. In-depth analysis of $\boldsymbol{\pi}$ selection

Rethinking the diffusion process. Mainstream diffusion models perturb the entire image with Gaussian noise and then perform pixel-wise reconstruction, aiming to handle noise corruption and recover high-quality information in parallel. For global degradations (e.g., low-light, noise), this approach achieves favorable performance by leveraging the known distribution of noise to guide the restoration of missing details. However, for mask-based degradations (e.g., rain, snow), only the degraded regions require restoration, while unaffected areas remain nearly identical to high-quality images. This approach introduces additional task complexity, which not only enables recovery of degraded regions, but also simultaneously compromises the quality of intact areas through redundant reconstruction. Moreover, severely degraded regions (with limited preserved information) benefit from enhanced noise perturbation to facilitate reconstruction, while mildly degraded regions require noise suppression to retain valid information. Drawing from the above analyses, $\boldsymbol{\pi}$ should possess weighted masking properties, effectively equivalent to the image residual $\boldsymbol{\pi} = \mathbf{x}_T - \mathbf{x}_0$.

Power analysis. Eq. (C.22) reveals that the diffusion process is determined by two terms given the final state $\mathbf{x}_T = \boldsymbol{\mu}$. The power ratio between residual component and noise component at pixel i, j can be defined as residual-to-noise ratio $R(t, i, j)$:

$$R(t,i,j) = \frac{(\mathbf{x}_T(i,j) - \mathbf{x}_0(i,j))^2}{2\boldsymbol{\pi}^2(i,j)\lambda}\frac{(\frac{\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})})^2}{\frac{\sinh(\theta_{0:t})\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})}} = \frac{(\mathbf{x}_T(i,j) - \mathbf{x}_0(i,j))^2}{2\boldsymbol{\pi}^2(i,j)\lambda}\frac{\sinh(\theta_{t:T})}{\sinh(\theta_{0:t})\sinh(\theta_{0:T})}, \tag{D.1}$$

The first part is determined by predefined parameters $\boldsymbol{\pi}, \lambda$ given initial and final states. The second part is entirely determined by the sequence of θ_t values, which approaches infinity at time 0 and converges to infinitesimal at time T. If $\boldsymbol{\pi}$ is a globally predefined parameter, when pixel residual $(\mathbf{x}_T(i,j) - \mathbf{x}_0(i,j))$ approaches zero, $R(t,i,j) \to \infty$. In this context, the high-quality regions are disrupted by noise and cannot be perfectly reconstructed due to the predicted error. Besides, the refinement

of low-quality regions with varying degradation degrees is dominated by their respective residual magnitudes. To make the $R(t,i,j)$ smooth, we leverage the setting of $\boldsymbol{\pi} = \mathbf{x}_T - \mathbf{x}_0$, and obtain:

$$R(t,i,j) = R(t) = \frac{\sinh(\theta_{t:T})}{\sinh(\theta_{0:t})\sinh(\theta_{0:T})}. \tag{D.2}$$

Let us check the monotonic properties of $R(t)$ by its logarithm derivatives:

$$A(t) = \sinh(\theta_{t:T}), \qquad B(t) = \sinh(\theta_{0:t}), \qquad C = \sinh(\theta_{0:T}), \tag{D.3}$$

$$A'(t) = \cosh(\theta_{t:T}) \cdot \frac{d}{dt}\theta_{t:T} = -\theta_t \cosh(\theta_{t:T}), \tag{D.4}$$

$$B'(t) = \cosh(\theta_{0:t}) \cdot \frac{d}{dt}\theta_{0:t} = \theta_t \cosh(\theta_{t:T}), \tag{D.5}$$

$$\frac{dR(t)}{dt} = \frac{A'(t)B(t) - A(t)B'(t)}{B^2(t)C} = -\theta_t \frac{\cosh(\theta_{t:T})\sinh(\theta_{0:t}) + \sinh(\theta_{t:T})\cosh(\theta_{0:t})}{\sinh^2(\theta_{0:t})\sinh(\theta_{0:T})} = -\frac{\theta_t}{\sinh^2(\theta_{0:t})}. \tag{D.6}$$

For the typical condition that $\theta_t \geq 0$, and $\sinh^2(\theta_{t:T}) \geq 0$, we can conclude that $R(t)$ is a monotonically decreasing function starting from $R(0) \to \infty$ to $R(T) = 0$, as:

$$\frac{d}{dt}R(t) \leq 0. \tag{D.7}$$

It can be observed that if $\boldsymbol{\pi} = \mathbf{x}_T - \mathbf{x}_0$ is set, $R(t)$ decreases evenly for each pixel without being affected by image contents. Hence, we set $\boldsymbol{\pi}$ as residual component in Sec. 4.1.

E. Process of Reverse Inference

For simplicity, we use Θ_t and Σ_t to represent the coefficients in Eq. (C.24) and Eq. (C.25), respectively. We have:

$$\Theta_t \equiv \frac{\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})}, \quad \Sigma_t \equiv 2\lambda \frac{\sinh(\theta_{0:t})\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})} \tag{E.1}$$

DDPM Reverse Process. Leveraging the properties of Bayesian formula, we obtain:

$$p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0, \mathbf{x}_T) = \frac{p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0, \mathbf{x}_T)p(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_T)}{p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T)}, \tag{E.2}$$

$$\mathbf{x}_{t-1} = \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_{t-1} + \boldsymbol{\pi}\Sigma_{t-1}\epsilon_{t-1}, \tag{E.3}$$

$$\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_t + \boldsymbol{\pi}\Sigma_t\epsilon_t. \tag{E.4}$$

Eliminating the variable $\mathbf{x}_0$, we have:

$$\mathbf{x}_t = \boldsymbol{\mu} + \Theta_t \frac{\mathbf{x}_{t-1} - \boldsymbol{\mu} - \Sigma_{t-1}\epsilon_{t-1}}{\Theta_{t-1}} + \boldsymbol{\pi}\Sigma_t\epsilon_t \tag{E.5}$$

$$= \boldsymbol{\mu} + \frac{\Theta_t}{\Theta_{t-1}}(\mathbf{x}_{t-1} - \boldsymbol{\mu}) + \boldsymbol{\pi}\sqrt{\Sigma_t^2 - \frac{\Theta_t^2}{\Theta_{t-1}^2}\Sigma_{t-1}^2}\epsilon \tag{E.6}$$

Back to Eq. (E.2), we have:

$$\begin{aligned}
&\log p(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T) = \log p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0, \mathbf{x}_T) + \log p(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_T) - \log p(\mathbf{x}_t|\mathbf{x}_0, \mathbf{x}_T) \\
&\propto -\frac{1}{2\boldsymbol{\pi}^2}\left[\frac{(\mathbf{x}_t - \boldsymbol{\mu} - \frac{\Theta_t}{\Theta_{t-1}}(\mathbf{x}_{t-1} - \boldsymbol{\mu}))^2}{\Sigma_t^2 - \frac{\Theta_t^2}{\Theta_{t-1}^2}\Sigma_{t-1}^2} + \frac{(\mathbf{x}_{t-1} - \boldsymbol{\mu} - (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_{t-1})^2}{\Sigma_{t-1}^2} - \frac{(\mathbf{x}_t - \boldsymbol{\mu} - (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_t)^2}{\Sigma_t^2}\right] \\
&= -\frac{1}{2\boldsymbol{\pi}^2}\left[\frac{(\mathbf{x}_{t-1} - \boldsymbol{\mu} - \frac{\Theta_{t-1}}{\Theta_t}(\mathbf{x}_t - \boldsymbol{\mu}))^2}{\frac{\Theta_{t-1}^2}{\Theta_t^2}\Sigma_t^2 - \Sigma_{t-1}^2} + \frac{(\mathbf{x}_{t-1} - \boldsymbol{\mu} - (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_{t-1})^2}{\Sigma_{t-1}^2} + C\right] \\
&= -\frac{1}{2\boldsymbol{\pi}^2}\left[\frac{(\mathbf{x}_{t-1}^2 - 2(\boldsymbol{\mu} + \frac{\Theta_{t-1}}{\Theta_t}(\mathbf{x}_t - \boldsymbol{\mu}))\mathbf{x}_{t-1} + (\boldsymbol{\mu} + \frac{\Theta_{t-1}}{\Theta_t}(\mathbf{x}_t - \boldsymbol{\mu}))^2}{\frac{\Theta_{t-1}^2}{\Theta_t^2}\Sigma_t^2 - \Sigma_{t-1}^2}\right. \\
&\qquad\qquad \left. + \frac{(\mathbf{x}_{t-1}^2 - 2(\boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_{t-1})\mathbf{x}_{t-1} + (\boldsymbol{\mu} + (\mathbf{x}_0 - \boldsymbol{\mu})\Theta_{t-1})^2}{\Sigma_{t-1}^2} + C\right]
\end{aligned} \tag{E.7}$$

Furthermore, all the terms not related to $\mathbf{x}_{t-1}$ are categorized as C:

$$\begin{aligned}\log p(\mathbf{x}_{t-1}|\mathbf{x}_0,\mathbf{x}_t,\mathbf{x}_T) = -\frac{1}{2\boldsymbol{\pi}^2}\Bigg[&(\frac{1}{\frac{\Theta_{t-1}^2}{\Theta_t^2}\Sigma_t^2-\Sigma_{t-1}^2}+\frac{1}{\Sigma_{t-1}^2})\mathbf{x}_{t-1}^2 - 2\big[\Sigma_{t-1}^2(\boldsymbol{\mu}+\frac{\Theta_{t-1}}{\Theta_t}(\mathbf{x}_t-\boldsymbol{\mu}))\\ &+(\frac{\Theta_{t-1}^2}{\Theta_t^2}\Sigma_t^2-\Sigma_{t-1}^2)(\boldsymbol{\mu}+(\mathbf{x}_0-\boldsymbol{\mu})\Theta_{t-1})\big]\mathbf{x}_{t-1}+C.\Bigg]\end{aligned} \tag{E.8}$$

We can reformulate the Eq. (E.8) in Gaussian distribution format:

$$Var[\mathbf{x}_{t-1}] = \boldsymbol{\pi}^2(\frac{1}{\frac{\Theta_{t-1}^2}{\Theta_t^2}\Sigma_t^2-\Sigma_{t-1}^2}+\frac{1}{\Sigma_{t-1}^2})^{-1} = \boldsymbol{\pi}^2\frac{\Sigma_{t-1}^2(\Theta_{t-1}^2\Sigma_t^2-\Theta_t^2\Sigma_{t-1}^2)}{\Theta_{t-1}^2\Sigma_t^2} \tag{E.9}$$

$$\begin{aligned}E[\mathbf{x}_{t-1}] &= \big[\Sigma_{t-1}^2(\boldsymbol{\mu}+\frac{\Theta_{t-1}}{\Theta_t}(\mathbf{x}_t-\boldsymbol{\mu}))+(\frac{\Theta_{t-1}^2}{\Theta_t^2}\Sigma_t^2-\Sigma_{t-1}^2)(\boldsymbol{\mu}+(\mathbf{x}_0-\boldsymbol{\mu})\Theta_{t-1})\big]\cdot\frac{\Sigma_{t-1}^2(\Theta_{t-1}^2\Sigma_t^2-\Theta_t^2\Sigma_{t-1}^2)}{\Theta_{t-1}^2\Sigma_t^2}\\ &= \frac{\Sigma_{t-1}^2(\Theta_{t-1}^2\Sigma_t^2-\Theta_t^2\Sigma_{t-1}^2)}{\Theta_t^2}[\boldsymbol{\mu}+(\mathbf{x}_0-\boldsymbol{\mu})\Theta_{t-1}]+\frac{\Sigma_{t-1}^4(\Theta_{t-1}^2\Sigma_t^2-\Theta_t^2\Sigma_{t-1}^2)}{\Theta_{t-1}\Sigma_t\Theta_t}\boldsymbol{\pi}\epsilon_t\end{aligned} \tag{E.10}$$

DDIM Reverse Process. A common forward process in our framework can be determined as follows:

$$\mathbf{x}_{t-1} = \boldsymbol{\mu}+(\mathbf{x}_0-\boldsymbol{\mu})\Theta_{t-1}+\boldsymbol{\pi}\Sigma_{t-1}\epsilon_{t-1}, \tag{E.11}$$

$$\mathbf{x}_t = \boldsymbol{\mu}+(\mathbf{x}_0-\boldsymbol{\mu})\Theta_t+\boldsymbol{\pi}\Sigma_t\epsilon_t. \tag{E.12}$$

We assume the reverse process follows a Gaussian distribution:

$$\begin{aligned}\mathbf{x}_{t-1} &= \kappa_t\mathbf{x}_t+\eta_t\boldsymbol{\mu}+\gamma_t\mathbf{x}_0+\dot{\sigma}_t\boldsymbol{\pi}\epsilon_t\\ &= \kappa_t(\boldsymbol{\mu}+(\mathbf{x}_0-\boldsymbol{\mu})\Theta_t+\boldsymbol{\pi}\Sigma_t\epsilon_t)+\eta_t\boldsymbol{\mu}+\gamma_t\mathbf{x}_0+\dot{\sigma}_t\boldsymbol{\pi}\epsilon_t\\ &= (\kappa_t+\eta_t-\kappa_t\Theta_t)\boldsymbol{\mu}+(\kappa_t\Theta_t+\gamma_t)\mathbf{x}_0+\boldsymbol{\pi}(\kappa_t^2\Sigma_t^2+{\dot{\sigma}_t}^2)^{\frac{1}{2}}\epsilon_t,\end{aligned} \tag{E.13}$$

we have:

$$\kappa_t+\eta_t-\kappa_t\Theta_t = 1-\Theta_{t-1}, \tag{E.14}$$

$$\kappa_t\Theta_t+\gamma_t = \Theta_{t-1}, \tag{E.15}$$

$$\Sigma_{t-1}^2 = \kappa_t^2\Sigma_t^2+{\dot{\sigma}_t}^2. \tag{E.16}$$

By setting $\dot{\sigma}_t = 0$:

$$\begin{aligned}\kappa_t = \frac{\Sigma_{t-1}}{\Sigma_t}, \gamma_t &= \Theta_{t-1}-\Theta_t\frac{\Sigma_{t-1}}{\Sigma_t}\\ \eta_t &= 1-\Theta_{t-1}-(1-\Theta_t)\frac{\Sigma_{t-1}}{\Sigma_t},\end{aligned} \tag{E.17}$$

substituting into Eq. (E.13):

$$\begin{aligned}\mathbf{x}_{t-1} &= \frac{\Sigma_{t-1}}{\Sigma_t}\mathbf{x}_t+(1-\Theta_{t-1}-(1-\Theta_t)\frac{\Sigma_{t-1}}{\Sigma_t})\boldsymbol{\mu}+(\Theta_{t-1}-\Theta_t\frac{\Sigma_{t-1}}{\Sigma_t})\mathbf{x}_0\\ &= \frac{\Sigma_{t-1}}{\Sigma_t}\mathbf{x}_t+(1-\frac{\Sigma_{t-1}}{\Sigma_t}-(\Theta_{t-1}-\Theta_t\frac{\Sigma_{t-1}}{\Sigma_t}))\boldsymbol{\mu}+(\Theta_{t-1}-\Theta_t\frac{\Sigma_{t-1}}{\Sigma_t})\mathbf{x}_0\\ &= \frac{\Sigma_{t-1}}{\Sigma_t}\mathbf{x}_t+(1-\frac{\Sigma_{t-1}}{\Sigma_t})\boldsymbol{\mu}+(\Theta_{t-1}-\Theta_t\frac{\Sigma_{t-1}}{\Sigma_t})(\mathbf{x}_0-\boldsymbol{\mu})\\ &= \frac{\Sigma_{t-1}}{\Sigma_t}\mathbf{x}_t+(1-\frac{\Sigma_{t-1}}{\Sigma_t})\boldsymbol{\mu}+(\Theta_{t-1}-\Theta_t\frac{\Sigma_{t-1}}{\Sigma_t})(\frac{\mathbf{x}_t-\boldsymbol{\mu}-\boldsymbol{\pi}\Sigma_t\epsilon_t}{\Theta_t})\\ &= \boldsymbol{\mu}+\frac{\Theta_{t-1}}{\Theta_t}(\mathbf{x}_t-\boldsymbol{\mu})-\boldsymbol{\pi}(\frac{\Theta_{t-1}}{\Theta_t}\Sigma_t-\Sigma_{t-1})\epsilon_t\end{aligned} \tag{E.18}$$

This concludes the derivations in Sec. 4.2.

F. Connections Among Existing Diffusion Bridge

Suppose the high-quality image $\mathbf{x}$ is sampled from the data distribution $p_{HQ}(\boldsymbol{x})$ and the paired degraded image $\boldsymbol{\mu}$ is sampled from prior distribution $p_{LQ}(\boldsymbol{x})$. We redefine the generalized meaning-reverting process as:

$$d\mathbf{x}_t = \theta_t(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t d\omega_t, \tag{F.1}$$

where $\boldsymbol{\pi}$ is a predefined value. By applying the Doob's h-transform, we can establish the bridge SDE that connects the paired distribution under a fixed drift-to-diffusion coefficient ratio $\lambda = \sigma_t^2/(2\theta_t)$:

$$d\mathbf{x}_t = \theta_t \coth(\overline{\theta}_{t:T})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sqrt{2\boldsymbol{\pi}^2\lambda\theta_t}d\omega_t, \tag{F.2}$$

F.1. Connections to Variance-Exploding (VE) and Variance-Preserving (VP) SDEs

SMLD [62] primarily introduces two mainstream diffusion formulations, namely VP and VE. For a given generalized OU process in Eq. (F.1), there exists relationships:

$$\begin{aligned}
\lim_{\theta_t\to 0}^{\boldsymbol{\pi}=1} \text{Eq. (F.1)} &= \lim_{\theta_t\to 0}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = \theta_t(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t d\omega_t\} \\
&= \lim_{\theta_t\to 0}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = \sigma_t d\omega_t\} \\
&= \text{VE},
\end{aligned} \tag{F.3}$$

where σ_t can be any noise schedule. Besides, we have:

$$\begin{aligned}
\lim_{\boldsymbol{\mu}\to 0,\theta_t\to\sigma_t^2}^{\boldsymbol{\pi}=1} \text{Eq. (F.1)} &= \lim_{\boldsymbol{\mu}\to 0,\theta_t\to\sigma_t^2}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = \theta_t(\boldsymbol{\mu} - \mathbf{x}_t)dt + \boldsymbol{\pi}\sigma_t d\omega_t\} \\
&= \lim_{\boldsymbol{\mu}\to 0,\theta_t\to\sigma_t^2}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = \theta_t\boldsymbol{\mu}dt - \theta_t\mathbf{x}_t dt + \sigma_t d\omega_t\} \\
&= \lim_{\boldsymbol{\mu}\to 0,\theta_t\to\sigma_t^2}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = -\frac{1}{2}\sigma_t^2\mathbf{x}_t dt + \sigma_t d\omega_t\} \\
&= \text{VP},
\end{aligned} \tag{F.4}$$

where we set the $\theta_t \to \sigma_t^2$, implying $\lambda = \frac{1}{2}$. On this basis, DDBM [83] further extends such diffusion configuration to bridge models, which are also special cases of our formulation under specific configurations:

$$\begin{aligned}
\lim_{\theta_t\to 0,\sigma_t^2\to C}^{\boldsymbol{\pi}=1} \text{Eq. (F.2)} &= \lim_{\theta_t\to 0,\sigma_t^2\to C}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = \theta_t \coth(\overline{\theta}_{t:T})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sqrt{2\pi^2\lambda\theta_t}d\omega_t\} \\
&= \lim_{\theta_t\to 0,\sigma_t^2\to C}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = \frac{\sigma_t^2}{\sigma_T^2 - \sigma_t^2}(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sigma_t d\omega_t\} \\
&= \text{VE Bridge},
\end{aligned} \tag{F.5}$$

where C denotes a constant and $\lambda = \frac{\sigma_t^2}{2\theta_t} \to \infty$. We utilize the following approximation:

$$\lim_{\theta_t\to 0} \theta_t \coth(\overline{\theta}_{t:T}) = \theta \coth(\theta(T-t)) = \frac{1}{T-t}. \tag{F.6}$$

VP bridge drives terminate state towards a zero-mean Gaussian distribution, which satisfies:

$$\begin{aligned}
\lim_{\boldsymbol{\mu}\to 0,\theta_t\to\sigma_t^2}^{\boldsymbol{\pi}=1} \text{Eq. (F.2)} &= \lim_{\boldsymbol{\mu}\to 0,\theta_t\to\sigma_t^2}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = \theta_t \coth(\overline{\theta}_{t:T})(\boldsymbol{\mu} - \mathbf{x}_t)dt + \sqrt{2\pi^2\lambda\theta_t}d\omega_t\} \\
&= \lim_{\boldsymbol{\mu}\to 0,\theta_t\to\sigma_t^2}^{\boldsymbol{\pi}=1}\{d\mathbf{x}_t = -\sigma_t^2 \coth(\overline{\sigma}_{t:T}^2)\mathbf{x}_t dt + \sigma_t d\omega_t\} \\
&= \text{VP Bridge},
\end{aligned} \tag{F.7}$$

F.2. Connections to Brownian Bridge SDEs

Brownian bridge is a fundamental architecture for diffusion model, which are widely adopted in BBDM [29], I^{2}SB [34]. By setting $\theta_t \to 0$ with condition $2\lambda\theta_t = 1 = \sigma_t^2$, we can derive the Brownian Bridge as formulated below:

$$\begin{aligned}\lim_{\theta_t\to 0,\sigma_t^2\to 1}^{\boldsymbol{\pi}=1} \text{Eq. (F.2)} &= \lim_{\theta_t\to 0,\sigma_t^2\to 1}^{\boldsymbol{\pi}=1} \{d\mathbf{x}_t = \theta_t \coth(\bar{\theta}_{t:T})(\boldsymbol{\mu}-\mathbf{x}_t)dt + \sqrt{2\pi^2\lambda\theta_t}d\omega_t\} \\ &= \lim_{\theta_t\to 0,\sigma_t^2\to 1}^{\boldsymbol{\pi}=1} \{d\mathbf{x}_t = \frac{\boldsymbol{\mu}-\mathbf{x}_t}{T-t}dt + d\omega_t\} \\ &= \text{Brownian Bridge},\end{aligned} \tag{F.8}$$

where the corresponding expectation and variance are:

$$\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_0-\boldsymbol{\mu})(1-\frac{t}{T}) + \int_0^t \frac{T-t}{T-s}dw_s, \tag{F.9}$$

$$E[\mathbf{x}_t] = \boldsymbol{\mu} + (\mathbf{x}_0-\boldsymbol{\mu})(1-\frac{t}{T}), \tag{F.10}$$

$$Var[\mathbf{x}_t] = t(1-\frac{t}{T}), \tag{F.11}$$

F.3. Connections to Flow Matching

Flow-based generative models [32, 37] design a deterministic probability path that linearly interpolates between a prior and the data distribution, and then directly learn a time-dependent vector field whose integral trajectories realize this path. By discarding the stochastic noise ($\sigma_t = 0$) and adopting the Brownian bridge configuration, Eq. (F.2) can be transformed into:

$$\begin{aligned}\lim_{\theta_t\to 0}^{\boldsymbol{\pi}=0} \text{Eq. (F.2)} &= \lim_{\theta_t\to 0}^{\boldsymbol{\pi}=0} \{d\mathbf{x}_t = \theta_t \coth(\bar{\theta}_{t:T})(\boldsymbol{\mu}-\mathbf{x}_t)dt + \sqrt{2\pi^2\lambda\theta_t}d\omega_t\} \\ &= \lim_{\theta_t\to 0}^{\boldsymbol{\pi}=0} \{d\mathbf{x}_t = \frac{\boldsymbol{\mu}-\mathbf{x}_t}{T-t}dt\} \\ &= \text{Flow Matching},\end{aligned} \tag{F.12}$$

whose trajectories satisfy:

$$\mathbf{x}_t = \boldsymbol{\mu} + (\mathbf{x}_0-\boldsymbol{\mu})(1-\frac{t}{T}). \tag{F.13}$$

F.4. Connections to OU Bridge SDEs

Eq. (F.2) can be transformed into naive OU bridge [73] by setting $\pi = 1$ to recover global noise perturbation:

$$\begin{aligned}\lim_{\theta_t,\lambda}^{\boldsymbol{\pi}=1} \text{Eq. (F.2)} &= \lim_{\theta_t,\lambda}^{\boldsymbol{\pi}=1} \{d\mathbf{x}_t = \theta_t \coth(\bar{\theta}_{t:T})(\boldsymbol{\mu}-\mathbf{x}_t)dt + \sqrt{2\pi^2\lambda\theta_t}d\omega_t\} \\ &= \lim_{\theta_t,\lambda}^{\boldsymbol{\pi}=1} \{d\mathbf{x}_t = \frac{\boldsymbol{\mu}-\mathbf{x}_t}{T-t}dt\} \\ &= \text{OU Bridge},\end{aligned} \tag{F.14}$$

F.5. Connections to Stochastic Interpolants

Stochastic interpolants [1] define a unified framework for flows and diffusions, which can be expressed as:

$$\mathbf{x}_t = I(t,\mathbf{x}_0,\mathbf{x}_T) + \gamma(t)z, t \in [0,T], \tag{F.15}$$

whose boundary conditions are $I(0,\mathbf{x}_0,\mathbf{x}_T) = \mathbf{x}_0$ and $I(T,\mathbf{x}_0,\mathbf{x}_T) = \mathbf{x}_T$. Eq. (F.2) describes our probability path as:

$$E[\mathbf{x}_t] = \boldsymbol{\mu} + (\mathbf{x}_0-\boldsymbol{\mu})\frac{\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})}, \tag{F.16}$$

$$Var[\mathbf{x}_t] = 2\boldsymbol{\pi}^2\lambda\frac{\sinh(\theta_{0:t})\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})}. \tag{F.17}$$

Hence, the above process can be regarded as stochastic interpolants. The derivative of $I(t, \mathbf{x}_0, \mathbf{x}_T)$ to time t is fixed as:

$$\partial_t I(t, \mathbf{x}_0, \mathbf{x}_T) = \partial_t \frac{\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})}(\mathbf{x}_0 - \boldsymbol{\mu}), \tag{F.18}$$

and $\gamma(t)$ with boundary conditions $\gamma(0) = \gamma(T) = 0$ is:

$$\gamma(t)^2 = 2\boldsymbol{\pi}^2\lambda \frac{\sinh(\theta_{0:t})\sinh(\theta_{t:T})}{\sinh(\theta_{0:T})}. \tag{F.19}$$

These relationships are summarized in Tab. 1 in Sec. 4.3.

G. Training Objective

Proposition 3 *Let $\mathbf{x}_t$ be a finite random variable described by the given residual diffusion bridge in Eq. (F.2). For a fixed final state $\mathbf{x}_T = \boldsymbol{\mu}$. the expectation of log-likelihood $\mathbb{E}_{p(\mathbf{x}_0)}[\log p_\theta(\mathbf{x}_0|\boldsymbol{\mu})]$ possesses an Evidence Lower Bound (ELBO):*

$$ELBO = \mathbb{E}_{p(\mathbf{x}_0)}\left[\mathbb{E}_{p(\mathbf{x}_1|\mathbf{x}_0,\boldsymbol{\mu})}[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1, \mathbf{x}_T)] - \sum_{t>1}\mathbb{E}_{p(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\mu})}[D_{KL}(p(\mathbf{x}_{t-1}|\mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T))\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)]\right] \tag{G.1}$$

Assuming $p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_T)$ follows a Gaussian distribution with a constant variance $\mathcal{N}(\boldsymbol{\mu}_{\theta,t-1}, \sigma^2_{\theta,t-1}I)$, maximizing the ELBO is equivalent to minimizing:

$$\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t,\mathbf{x}_T}\left[\frac{1}{2\sigma^2_{\theta,t-1}}\|\boldsymbol{\mu}_{t-1} - \boldsymbol{\mu}_{\theta,t-1}\|^2\right] \tag{G.2}$$

where μ_{t-1} is the expectation at time $t-1$ and $\mu_{\theta,t-1}$ is predicted by a neural network parameterized by θ.

Proof. For the conditional marginal likelihood of the data $\mathbf{x}_0$, we have

$$p_\theta(\mathbf{x}_0|\boldsymbol{\mu}) = \int p_\theta(\mathbf{x}_{0:T}|\boldsymbol{\mu})d\mathbf{x}_{1:T} = \int \frac{p_\theta(\mathbf{x}_{0:T}|\boldsymbol{\mu})}{p(\mathbf{x}_{1:T}|\mathbf{x}_0, \boldsymbol{\mu})}p(\mathbf{x}_{1:T}|\mathbf{x}_0, \boldsymbol{\mu})d\mathbf{x}_{1:T} \tag{G.3}$$

To maximize Eq. (G.3), we leverage the property of Jensen's inequality:

$$\log p_\theta(\mathbf{x}_0|\boldsymbol{\mu}) \geq \mathbb{E}_{p(\mathbf{x}_{1:T}|\mathbf{x}_0,\boldsymbol{\mu})}\left[\log \frac{p_\theta(\mathbf{x}_{0:T}|\boldsymbol{\mu})}{p(\mathbf{x}_{1:T}|\mathbf{x}_0, \boldsymbol{\mu})}\right] = \mathbb{E}\left[\log p_\theta(\mathbf{x}_T|\boldsymbol{\mu}) + \log \frac{p_\theta(\mathbf{x}_{0:T-1}|\boldsymbol{\mu})}{p(\mathbf{x}_{1:T}|\mathbf{x}_0, \boldsymbol{\mu})}\right] \tag{G.4}$$

$$= \mathbb{E}\left[\log p_\theta(\mathbf{x}_T|\boldsymbol{\mu}) + \sum_{t\geq 1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\mu})}{p(\mathbf{x}_t|\mathbf{x}_{t-1}, \boldsymbol{\mu})}\right] \tag{G.5}$$

$$= \mathbb{E}\left[\log p_\theta(\mathbf{x}_T|\boldsymbol{\mu}) + \sum_{t>1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\mu})}{p(\mathbf{x}_t|\mathbf{x}_{t-1}, \mathbf{x}_0, \boldsymbol{\mu})} + \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\mu})}{p(\mathbf{x}_1|\mathbf{x}_0, \boldsymbol{\mu})}\right] \tag{G.6}$$

$$= \mathbb{E}\left[\log p_\theta(\mathbf{x}_T|\boldsymbol{\mu}) + \sum_{t>1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\mu})}{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\mu})}\cdot\frac{p(\mathbf{x}_{t-1}|\mathbf{x}_0, \boldsymbol{\mu})}{p(\mathbf{x}_t|\mathbf{x}_0, \boldsymbol{\mu})} + \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\mu})}{p(\mathbf{x}_1|\mathbf{x}_0, \boldsymbol{\mu})}\right] \tag{G.7}$$

$$= \mathbb{E}\left[\log p_\theta(\mathbf{x}_T|\boldsymbol{\mu}) + \sum_{t>1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\mu})}{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\mu})} + \sum_{t>1}\log \frac{p(\mathbf{x}_{t-1}|\mathbf{x}_0, \boldsymbol{\mu})}{p(\mathbf{x}_t|\mathbf{x}_0, \boldsymbol{\mu})} + \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\mu})}{p(\mathbf{x}_1|\mathbf{x}_0, \boldsymbol{\mu})}\right] \tag{G.8}$$

$$= \mathbb{E}\left[\log p_\theta(\mathbf{x}_T|\boldsymbol{\mu}) + \sum_{t>1}\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\mu})}{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\mu})} + \log \frac{p(\mathbf{x}_1|\mathbf{x}_0, \boldsymbol{\mu})}{p(\mathbf{x}_T|\mathbf{x}_0, \boldsymbol{\mu})} + \log \frac{p_\theta(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\mu})}{p(\mathbf{x}_1|\mathbf{x}_0, \boldsymbol{\mu})}\right] \tag{G.9}$$

$$= \mathbb{E}\left[\log \frac{p_\theta(\mathbf{x}_T|\boldsymbol{\mu})}{p(\mathbf{x}_T|\mathbf{x}_0, \boldsymbol{\mu})}\right] + \sum_{t>1}\mathbb{E}\left[\log \frac{p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t, \boldsymbol{\mu})}{p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\mu})}\right] + \mathbb{E}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\mu})\right] \tag{G.10}$$

$$= \mathbb{E}_{p(\mathbf{x}_1|\mathbf{x}_0,\boldsymbol{\mu})}\left[\log p_\theta(\mathbf{x}_0|\mathbf{x}_1, \boldsymbol{\mu})\right] - \sum_{t>1}\mathbb{E}_{p(\mathbf{x}_t|\mathbf{x}_0,\boldsymbol{\mu})}\left[D_{KL}(p(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0, \boldsymbol{\mu})\|p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_T, \boldsymbol{\mu}))\right]. \tag{G.11}$$

Accordingly,

$$
\begin{aligned}
& D_{KL}\left(p\left(\mathbf{x}_{t-1} \mid \mathbf{x}_0, \mathbf{x}_t, \mathbf{x}_T\right) || p_\theta\left(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_T\right)\right) \\
= & \mathbb{E}_{p(\mathbf{x}_{t-1}|\mathbf{x}_0,\mathbf{x}_t,\mathbf{x}_T)}\left[\log \frac{\frac{1}{\sqrt{2\pi}\sigma_{t-1}} e^{-(x_{t-1}-\boldsymbol{\mu}_{t-1})^2/2\sigma_{t-1}^2}}{\frac{1}{\sqrt{2\pi}\sigma_{\theta,t-1}} e^{-(x_{t-1}-\boldsymbol{\mu}_{\theta,t-1})^2/2\sigma_{\theta,t-1}^2}}\right] \\
= & \mathbb{E}_{p(\mathbf{x}_{t-1}|\mathbf{x}_0,\mathbf{x}_t,\mathbf{x}_T)}\left[\log \sigma_{\theta,t-1} - \log \sigma_{t-1} - (x_{t-1}-\boldsymbol{\mu}_{t-1})^2/2\sigma_{t-1}^2 + (x_{t-1}-\boldsymbol{\mu}_{\theta,t-1})^2/2\sigma_{\theta,t-1}^2\right] \\
= & \log \sigma_{\theta,t-1} - \log \sigma_{t-1} - \frac{1}{2} + \frac{\sigma_{t-1}^2}{2\sigma_{\theta,t-1}^2} + \frac{(\boldsymbol{\mu}_{t-1}-\boldsymbol{\mu}_{\theta,t-1})^2}{2\sigma_{\theta,t-1}^2}
\end{aligned}
\tag{G.12}
$$

Ignoring unlearnable constant, the training objective that involves minimizing the negative ELBO is :

$$
\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t,\mathbf{x}_T}\left[\frac{1}{2\sigma_{\theta,t-1}^2}\|\boldsymbol{\mu}_{t-1} - \boldsymbol{\mu}_{\theta,t-1}\|^2\right]. \tag{G.13}
$$

By substituting Eq. (E.10) into Eq. (G.13) yields the equivalent loss:

$$
\mathcal{L} = \mathbb{E}_{t,\mathbf{x}_0,\mathbf{x}_t,\mathbf{x}_T}\left[C_\theta\|\boldsymbol{\pi}\epsilon_{t-1} - \boldsymbol{\pi}\epsilon_{\theta,t-1}\|^2\right]. \tag{G.14}
$$

Where C_θ are corresponding weights. This concludes the proof of the **Proposition 3** in Sec. 4.2.

H. More Experiments

H.1. Summary about the Datasets

We evaluate the proposed method on five natural image restoration tasks, including deraining, low-light enhancement, desnowing, dehazing, and deblurring. We select the most widely used datasets for each task, as summarized in Tab. S1.

Table S1. Summary of the image restoration datasets utilized in this paper.

Task	Dataset	Synthetic/Real	Train samples	Test samples
Deraining	DID [78]	Synthetic	-	1,200
	Rain13K [19]	Synthetic	13,711	-
	Rain_100 [72]	Synthetic	-	200
	DeRaindrop [51]	Real	861	307
	GT-Rain [4]	Real	26,125	2,100
	RealRain-1k [31]	Real	1792	448
Low-light Enhancement	LOL [68]	Real	485	15
	MEF [44]	Real	-	17
	VE-LOL-L [35]	Synthetic/Real	900/400	100/100
	NPE [66]	Real	-	8
	DICM [26]	Real	-	64
Desnowing	CSD [6]	Synthetic	8,000	2,000
	Snow100K-Real [38]	Real	-	1,329
Dehazing	SOTS [27]	Synthetic	-	500
	ITS_v2 [27]	Synthetic	13,990	-
	D-HAZY [10]	Synthetic	1,178	294
	NH-HAZE [3]	Real	-	55
	Dense-Haze [2]	Real	-	55
	NHRW [79]	Real	-	150
Deblur	GoPro [48]	Synthetic	2,103	1,111
	RealBlur [53]	Real	3,758	980

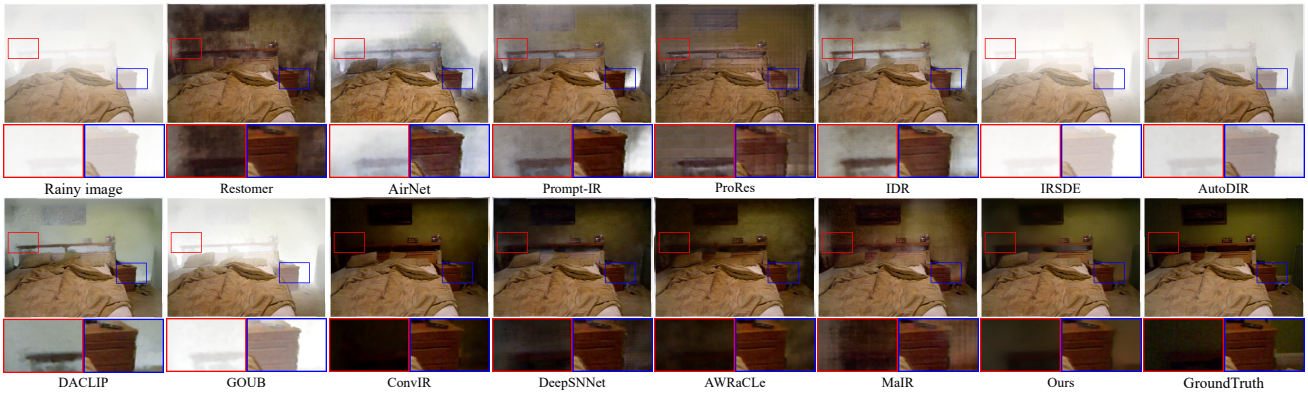


Figure S1. Visualization comparison with state-of-the-art methods on dehazing. Zoom in for best view.

Figure S2. Visualization comparison with state-of-the-art methods on deblurring. Zoom in for best view.

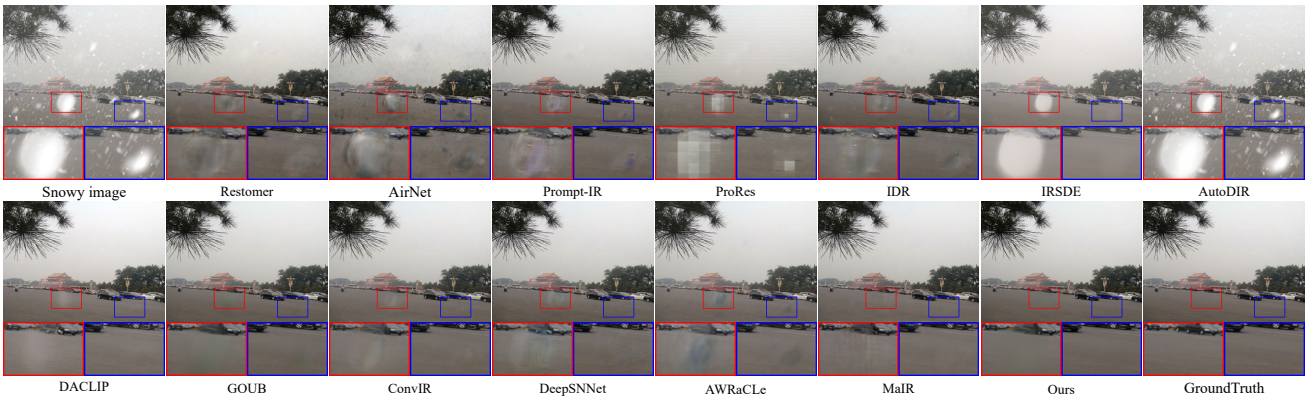


Figure S3. Visualization comparison with state-of-the-art methods on desnowing. Zoom in for best view.

H.2. More Visual Comparisons on Image Restoration

We show the visualization results of other degradation categories in Fig. S1, Fig. S2, Fig. S3, and Fig. S4, to further demonstrate our superiority. Evidently, our method generates more stable image samples with high fidelity than other universal image restoration methods. Benefiting from the adaptivity of residual bridge score matching, we achieve the outstanding reconstruction of the missing details and preserve undegraded regions well.

H.3. More Visual Comparisons on Real-world Scene Generalization

Known task generalization. We randomly select 20 samples for each task to conduct the non-reference assessment, as presented in Tab. 8. Furthermore, to fully demonstrate that our method can handle the real-world restoration tasks, we have generalized all well-optimized models to five known tasks within real-world scenarios. Visual comparisons are displayed in Fig. S5, Fig. S6, Fig. S7, Fig. S8, and Fig. S9, respectively. Clearly, our method produces the highest-quality restored images.

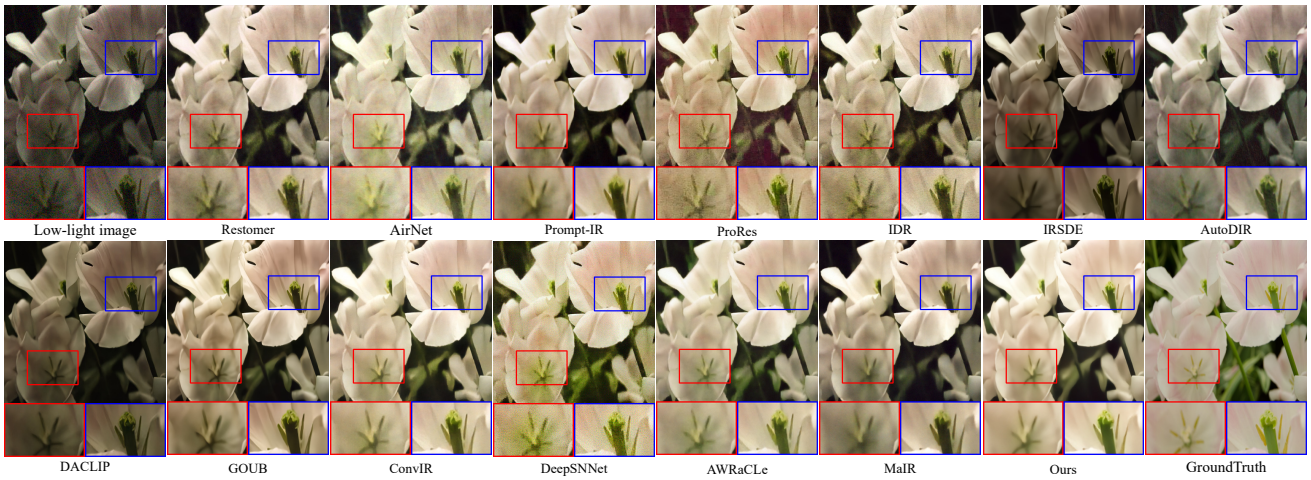


Figure S4. Visualization comparison with state-of-the-art methods on low-light enhancement. Zoom in for best view.

Figure S5. Visualization comparison of deblurring task in real-world scenarios. Zoom in for best view.

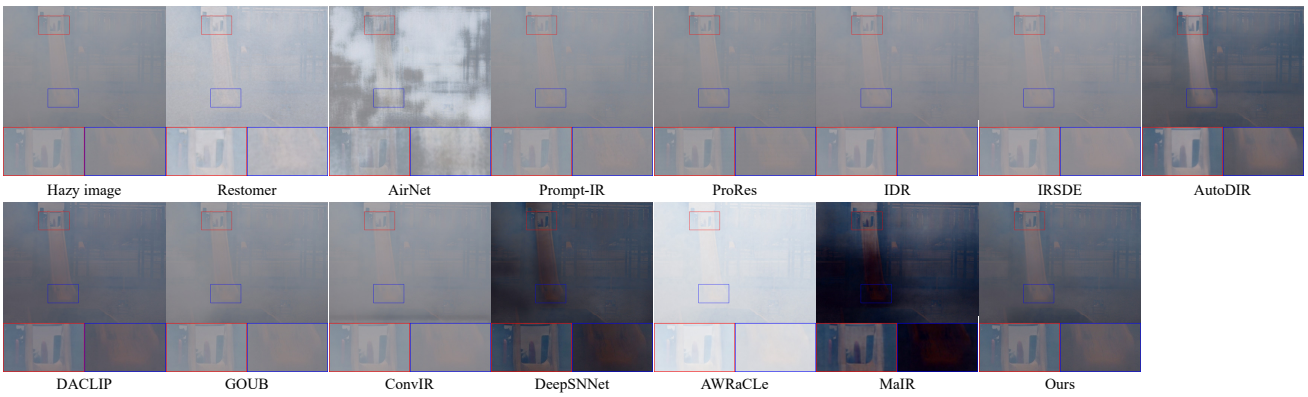


Figure S6. Visualization comparison of dehazing task in real-world scenarios. Zoom in for best view.

Unknown task generalization. Unknown task image restoration is performed on both POLED and TOLED [84]. Visual comparisons on the POLED dataset are provided in Fig. S10. The results show that our method can generalize to real-world scenes and achieve competitive visual results.

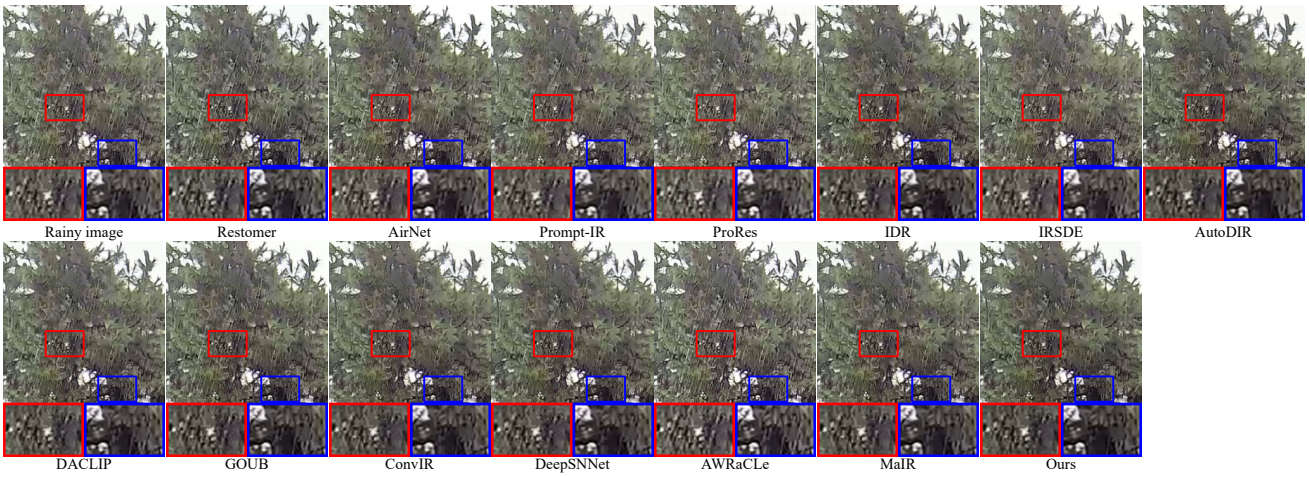


Figure S7. Visualization comparison of deraining task in real-world scenarios. Zoom in for best view.

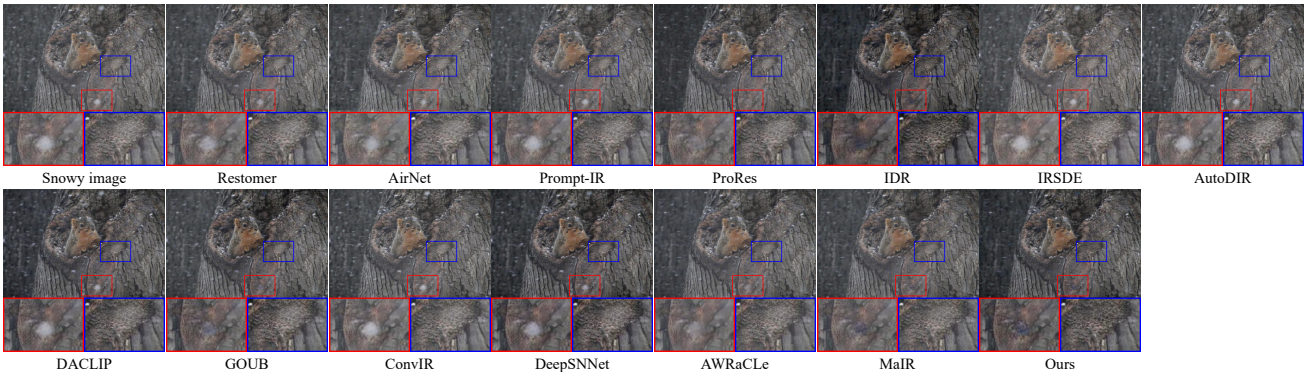


Figure S8. Visualization comparison of desnowing task in real-world scenarios. Zoom in for best view.

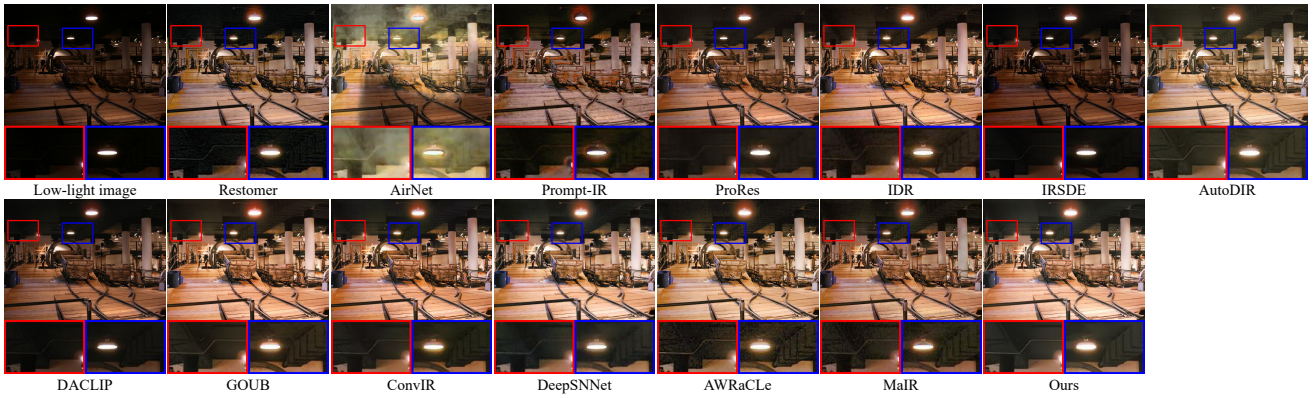


Figure S9. Visualization comparison of low-light enhancement task in real-world scenarios. Zoom in for best view.

H.4. More Visual Comparisons on Image Translation and Inpainting

To further show the visual advantages of our approach across tasks, we present additional comparisons for image translation (Fig. S11) and image inpainting (Fig. S12). In image translation, our method better preserves semantic and structural consistency, produces more faithful colors, sharper edges, and richer details. In image inpainting, it synthesizes textures and boundaries highly consistent with the surrounding context while avoiding oversmoothing and texture drift. Overall, our qualitative results show clearer details, stronger global consistency, and fewer visual artifacts than competing methods.

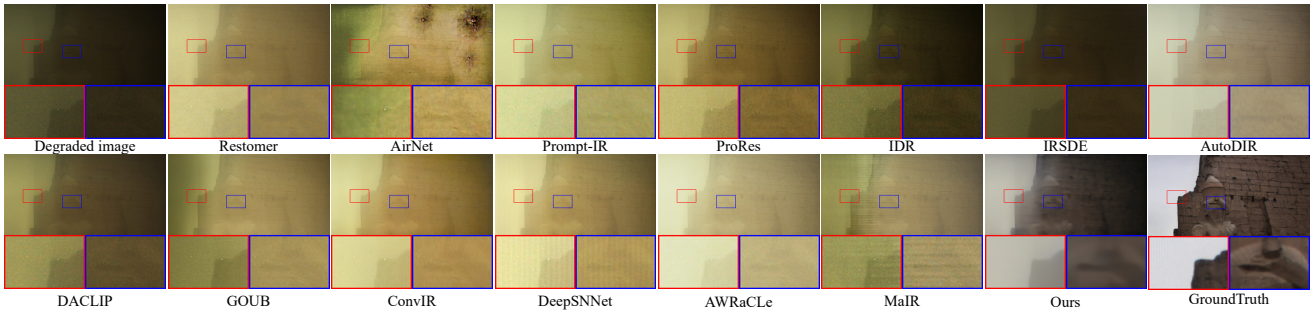


Figure S10. Visualization results of zero-shot generalization in real-world POLED dataset. Zoom in for best view.

Figure S11. Visualization results of image translation. Zoom in for best view.

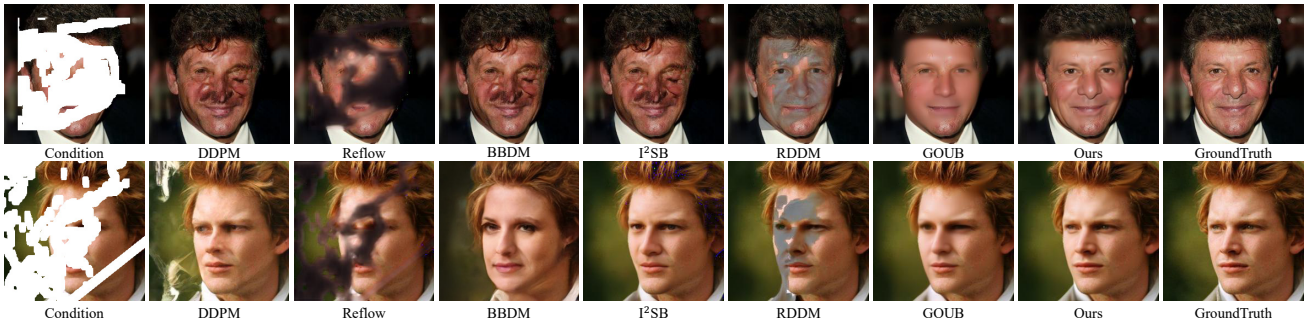


Figure S12. Visualization results of image inpainting. Zoom in for best view.

H.5. Efficiency Comparison

Our mixed dataset consists of images with resolutions ranging from 256 to 1024 pixels. Accordingly, we evaluate model efficiency under three representative resolution settings, as summarized in Tab. S2. Evidently, our methods are moderately efficient with reasonable resource consumption. Overall, RDBM strikes a balance between efficiency and performance.

I. Discussions, Limitations, and Future Work

Limitations and broader impact. The main challenge lies in fully exploring the connections between the data and prior distribution to modify the diffusion process. Although we have theoretically proposed a general and analytical formulation for diffusion bridge models, our core analysis assumes a fixed drift-to-diffusion coefficient ratio $\lambda = \sigma_t^2/(2\theta_t)$ to admit closed-form solutions of SDEs. In the fields of image restoration, translation and inpainting where the data and prior distributions share semantic or structural affinity, our method is highly flexible and robust with competitive performance. However, it may be sub-optimal when applied to the generative tasks, where the distributions lack direct correspondence. Despite current limitations, we believe our unified model offers a strong foundation for diffusion bridge models.

Future Work. Future work could be explored in several promising directions. (1) With the rise of high-resolution imagery (e.g., 4K, 8K), developing multi-dimensional latent diffusion bridge models is crucial to address the computational demands.

Table S2. Efficiency comparisons among universal methods. '-' means out of memmory.

Resolution	256×256			512×512			1024×1024		
Method	Mem.(G)	Time(s)	FPS	Mem.(G)	Time(s)	FPS	Mem.(G)	Time(s)	FPS
Restomer [76]	1.959	0.105	9.563	6.670	0.381	2.622	25.419	1.773	0.564
AirNet [28]	1.039	0.194	5.159	3.480	0.738	1.355	11.244	20.499	0.049
Prompt-IR [50]	2.544	0.111	8.981	7.255	0.399	2.508	26.005	1.845	0.542
ProRes [43]	2.027	0.318	3.149	2.514	0.766	1.305	6.025	1.715	0.583
IDR [80]	1.340	0.052	19.253	4.313	0.136	7.373	16.110	0.615	1.626
IRSDE [40]	1.554	5.017	0.199	2.743	18.493	0.054	9.997	72.289	0.014
AutoDIR [20]	7.023	6.266	0.160	11.021	11.986	0.083	—	—	—
DA-CLIP [41]	2.119	2.585	0.387	6.775	7.937	0.126	58.548	60.893	0.016
GOUB [73]	1.554	4.996	0.200	2.868	18.442	0.054	10.122	72.239	0.014
ConvIR [11]	0.708	0.035	28.570	1.184	0.055	18.020	2.809	0.192	5.202
DeepSNNet [12]	0.862	0.100	9.974	0.989	0.102	9.801	1.364	0.267	3.749
AWRaCLe [52]	1.929	0.101	9.922	4.264	0.354	2.822	13.608	1.374	0.728
MaIR [30]	2.091	1.297	0.771	6.593	4.744	0.211	24.593	18.029	0.055
RDBM-T	0.813	0.418	2.392	1.907	1.621	0.617	13.407	6.648	0.150
RDBM-S	0.825	0.431	2.322	1.921	1.648	0.607	13.421	6.775	0.148
RDBM-B	1.124	0.480	2.081	2.186	1.926	0.519	14.938	7.872	0.127
RDBM-L	1.150	0.504	1.986	2.307	1.982	0.505	15.059	8.135	0.123

(2) Exploring more efficient network architectures to reduce memory usage and enhance efficiency. (3) Expanding the model capacity and datasets to strengthen restoration performance and generalization. (4) Designing adaptive learning rate schedules or applying model distillation to reduce sampling steps and improve restoration quality.